Универзитет ,,Гоце Делчев”

во Штип

Скрипта по БИОИНФОРМАТИКА

Автор:

Доне Стојанов

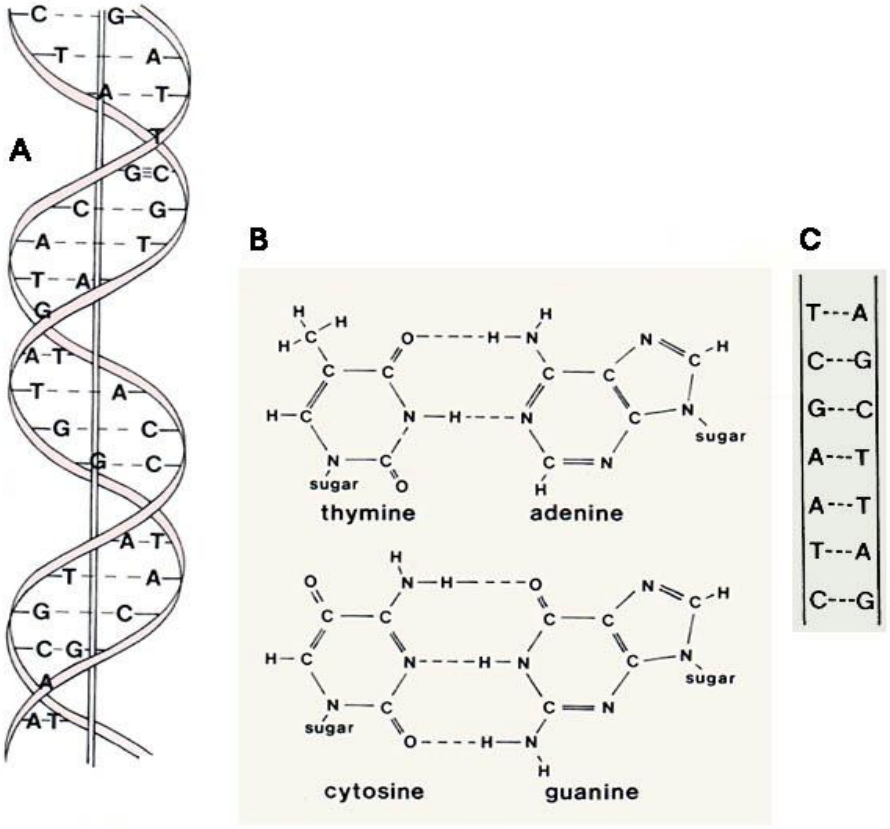
Предговор:

Ракописот претставува сеопфатен преглед на најчесто употребуваните алгоритми за порамнување и пребарување на генетски секвенци. Алгоритмите се објаснети на конкретни примери. На почеток е даден краток осврт на основните концепти од молекуларна биологија. На крај од ракописот, читателот се запознава со библиотеките на генетски податоци.

Содржина:

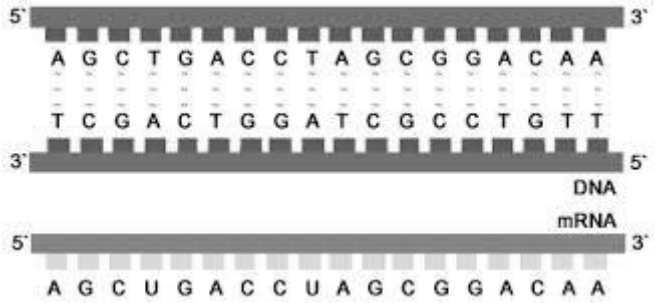
1. Вовед во молекуларната биологија

Деоксирибонуклеинската киселина е двојнонитна, издолжена и содржински комплементарна биолошка молекула, Слика 1.1 (А). ДНК молекулата содржи наследен генетски материјал. Секоја нитка е низа од четири основни нуклеотиди: А(*аденин*), G(*гуанин*), С(*цитозин*) и Т(*тимин*), Слика 1.1 (В). Фосфодиестер врските овозможуваат сериско поврзување на нуклеотидите по нитка. Важи правилото дека на секој нуклеотид од една нитка соодветствува нему комплементарен нуклеотид од спротивната нитка, Слика 1.1 (В). Аденин комплементарен нуклеотид е тиминот и обратно, додека цитозин комплементарен нуклеотид е гуанинот и обратно. Базните поврзувања се овозможени со двојни, односно тројни водородни врски, формирајќи стабилна двојно-нитна структура со дијаметар од 20 ангстроми ($1A=10^{-10}$ m), која има природен потенцијал за обнова во случај на оштета. Секој нуклеотид е изграден од деоксирибоза шеќер, фосфатна група и азотна база. Структурата на азотната база е различна за секој нуклеотид, додека зедничка е хемиската структурата на фосфатната група и шеќерот.



Слика 1.1. Приказ на ДНК молекула

Информацијата кодирана во ДНК се дели на *кодирачка* и *некодирачка*. Најголем процент од некодирачката ДНК припаѓа на кратките тандемски повторувања, кои во основа се повеќекратни повторувања на карок ДНК фрагмент како на пример СА фрагментот. Честотата на повторување се зема како главен индикатор за утврдување на генетскиот профил. Кодирачката ДНК содржина ја определуваат *гените – протеин кодирачки фрагменти*, чиј почеток и крај се определени преку *промотор* и *терминатор* секвенца. Имајќи ја во предвид варијабилноста на промотор секвенците, почетокот на генот го определува *промотор консензус* секвенца, која се наоѓа на 10 или 35 базни позиции на лево во однос на транскрипцискиот почеток. Со посредство на ензимот РНК полимераза, генот се транскибира во *гласник рибонуклеинска киселина* (*м-РНК*), Слика 1.2.

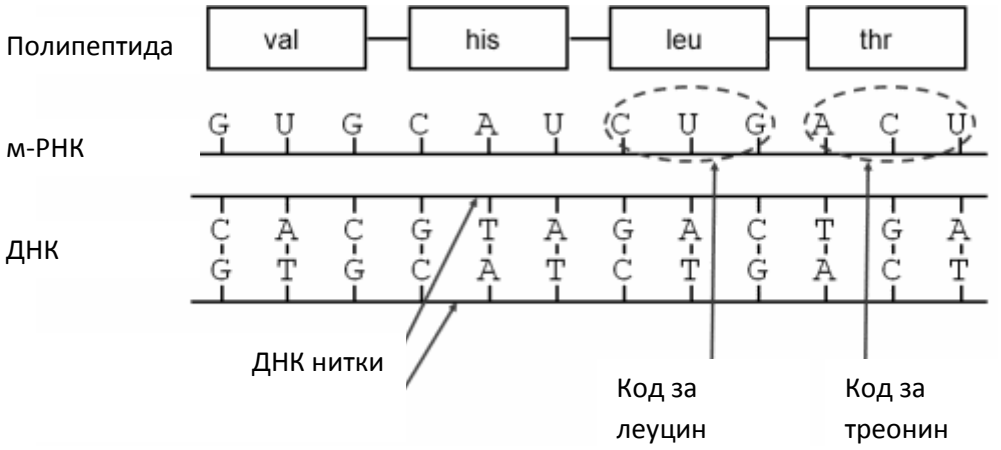


Слика 1.2. Транскрипција на ген

Во основа постојат три видови на РНК: *рибозомска РНК* (*р-РНК*), *транспортна РНК* (*т-РНК*) и *гласник РНК* (*м-РНК*). Ризбозомската РНК е структурна компонента на рибозомскиот комплекс и истата учествува во синтезата на протеини. Транспортната РНК е кратка молекула со должина од 70 до 90 базни парови. Истите делуваат како преносници на амино-киселините во фазата на синтеза на протеин. Информацијата содржана во гласник рибонуклеинската киселина служи како шаблон за синтеза на конкретен протеин. Протеините се вклучени во повеќето биолошки процеси и имат најразлични функционалности. Истите делуваат како: забрзувачи на биохемиските процеси, преносници на кислород и железо и одржувачи на структурата на клетката. Структурата на информациската РНК е комплементарна во однос на структурата на генот од ДНК нитката предмет на траскрипција, каде важи правилото дека аденин комплементарен рибонуклеотид е Урацилот (U), Слика 1.2. Кај РНК нуклеотидите, шеќерниот јаглероден атом $2^{'}$ е поврзан со хидроксилна – ОН група наместо само со водород како кај деоксирибозата.

Низата од три последователни нуклеотиди се нарекува *кодон*. Секој кодон соодветствува на една од двесетте протеински амино-киселини. Започнувајќи од *старт кодонот*, завршувајќи со *стоп кодонот*, секој кодон се преведува во конкретна амино-киселина, Слика 1.3, според *универзалниот генетски код*, Слика 1.4. Кодонот го препознава

конкретна т-РНК молекула, која ја носи и додава соодветната амино-киселина на крајот од растечката протеинска полипептида. Битно е да се напомене дека *примарниот м-РНК транскрипт* содржи *интрони* и *егзони*. Интроните се некодирачки РНК подсеквенци, додека егзоните се кодирачки РНК подсеквенци. Со отстранување на интроните се добива *конечен РНК транскрипт*, предмет на превод.



Слика 1.3. Превод на м-РНК

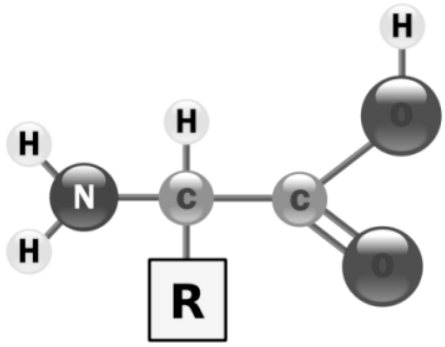
	U	C	A	G
U	UUU Phe UUC UUA Leu UUG	UCU UCC UCA Ser UCG	UAU Tyr UAC UAA Stop UAG	UGU Cys UGC UGA Stop UGG Trp
C	CUU CUC CUA Leu CUG	CCU CCC CCA Pro CCG	CAU His CAC CAA Gln CAG	CGU CGC CGA Arg CGG
A	AUU AUC Ile AUA AUG Met	ACU ACC ACA Thr ACG	AAU Asn AAC AAA Lys AAG	AGU Ser AGC AGA Arg AGG
G	GUU GUC GUA Val GUG	GCU GCC GCA Ala GCG	GAU Asp GAC GAA Glu GAG	GGU GGC GGA Gly GGG

Слика 1.4. Универзален генетски код

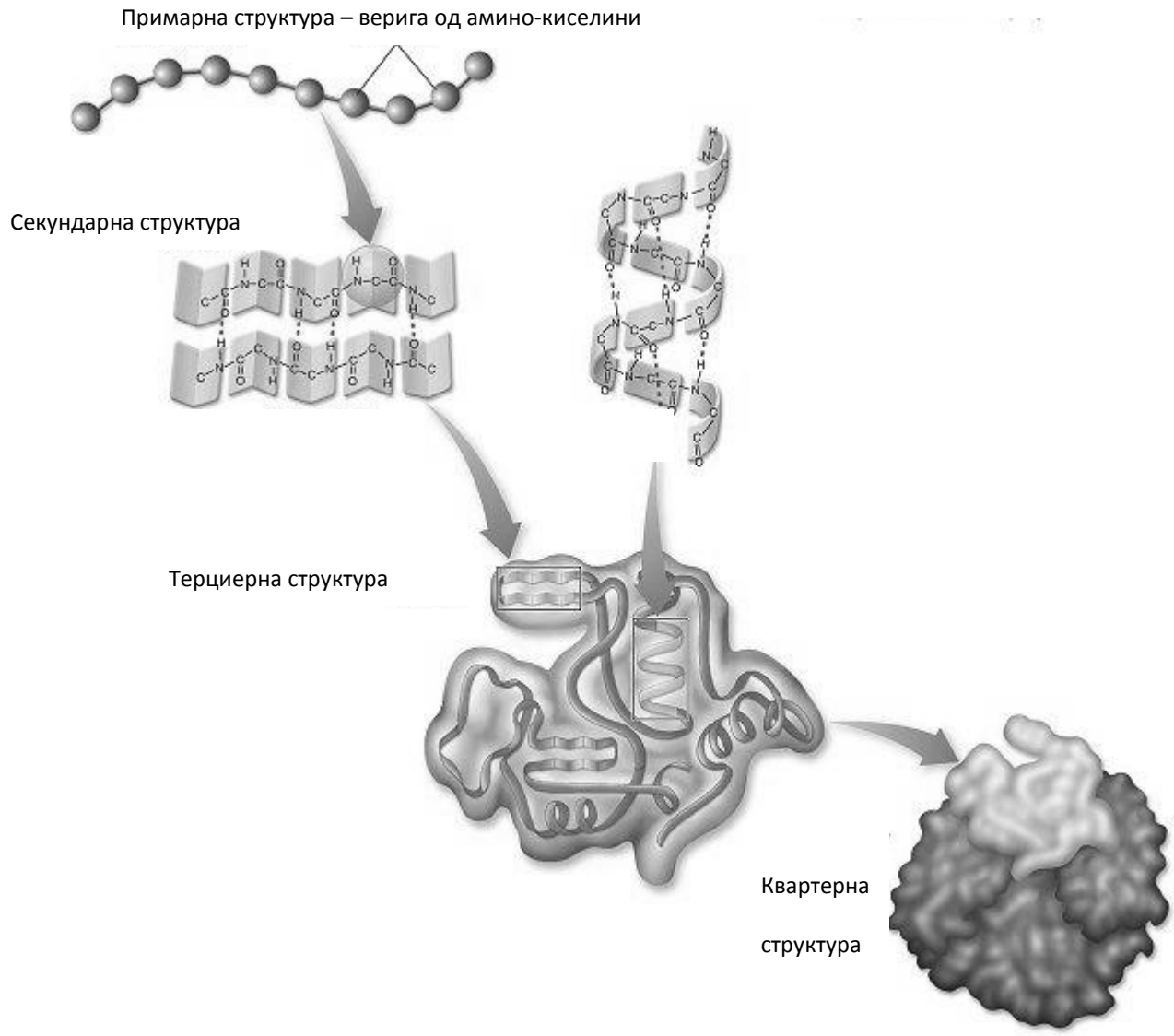
Досегашната фенотипска анализа укажува на варијации во структурата на гените. Последицата од промената на ген информацијата е синтеза на изменет протеин во фазата на превод. Синтезата на изменет протеин може да биде причина за различни функционални и структурни нарушувања. Драстични генетски промени се случуваат во услови на радијација и изложеност на надворешни хемиски фактори.

Базните промени во ниво на ген се познати како мутации. Можни се следниве мутации: замена на еден нуклеотид со друг, бришење на нуклеотид и додавање на нуклеотид. Во најлош случај, мутацијата: нетерминален триплет во стоп кодон ќе предизвика синтеза на скратен и најверојатно нефункционален протеин - потенцијална причина за функционално нарушување.

Резултатот од преводот на информацијата содржана во м-РНК е протеин. Протеините се линеарни вериги изградени од амино-кислеини. Во градбата на протеините се вклучени 20 различни амино-киселини, со различни биохемиски својства. Секоја амино-киселина е изградена од централен јаглероден атом, поврзан со амино група, карбоксилна група и остаток, чија структура е единствена за секоја амино-киселина, Слика 1.5. Поврзувањето во верига е овозможено со форирање на пептидни врски помеѓу амино и карбоксилната група. Фрагменти од протеините се извиткуваат во регуларни геометриски облици, како што се алфа хеликсите и бета рамнините, Слика 1.6. Просторното поврзување на секундарните извиткувања ја определува терциерната структура на протеините. Од друга страна пак, просторната структура на протеинот е во тесна релација со неговата функционалност. Протеини со слична просторна структура имаат и слична функционалност.



Слика 1.5. Структура на амино-киселина



Слика 1.6. Четири нивоа на структурна организација кај протеините

2. Порамнување на ДНК секвенци

2.1. Хомологија

Структурата на голем број протеини и нуклеотидни секвенци е позната и јавно достапна. И покрај тоа, малку се знае за повеќето гени. Невозможно е да се анализира секој ген поединечно. Ген со непозната функција, но со слична информациска содржина со ген, чија функција е експериментално утврдена, се очекува да има слична функционалност.Таквите гени се нарекуваат уште и *хомологни гени*. Процесот на утврдување на хомологија помеѓу две генетски секвенци се поедноставува со примена на компјутерски алгоритми.

Парцијалниот пристап за утврдување на сличност помеѓу две генетски секвенци се базира на пронаоѓање на заеднички совпаѓања. Резултатите би биле оптимални само во услови на базни субституции. Така на пример, секвенците: ACTTGT и AGTTGT се пример хомологни секвенци, со различен нуклеотид на положба 2. Споредувајќи ги базите на исти положби се утврдуваат совпаѓањата: A и TTGT, одделени со едно базно несповпаѓање. Биолошката интерпретација на резултатот укажува на базна субституција на положба 2.

Додавањата и бришењата на нуклеотиди се исто така генетски мутации, за кои претходниот концепт за утврдување на сличност е неприменлив. Пример секвенците: ACTTGT и ATTGT се исто така хомологни секвенци, кои се разликуваат по отсуството на нуклеотидот цитозин на позција 2 кај втората секвенца. Утврдувањето на сличноста врз основа на идентификација на совпаѓања, не ја отсликува реалната еволуциска поврзаност, што значи дека постои потреба од пофлексибилен, но и пософистициран пристап за утврдување на сличност помеѓу две генетски секвенци. Иделаното решение за пример секвенците би било: A_TTGT, кое што може да се интерпретира на два начини: кај втората секвенца нуклеотидот 2 е избришан или нуклеотидот цитозин е додаден на позиција 2 кај првата секвенца.

Некои субституции е поверојатно да се случат од други. Еволуциски гледано, веројатноста за субституција на една протеинска амино-киселина со друга е различна за различни субституциски парови. Добра мерика за порамнување не треба да се базира на додела на униформна казна за несовпаѓање. Казната за единечна субституција се пресметува како количник помеѓу веројатноста една амино-киселина да се замени со друга и вројатноста за случајна обсервација на истата амино-киселина во рамки на две неколерирани секвенци.

2.2. Утврдување на најдолг регион на совпаѓање

Најстариот метод за утврдување на сличност помеѓу две генетски секвенци е базиран на употреба на *матрица на точки*. За да се утврдат совпаѓањата помеѓу секвенците $a = \{a_i\}_{i=1}^{n}$ и $b = \{b_j\}_{j=1}^{m}$ се конструира матрица $[s_{i,j}]_{1 \le i \le n, 1 \le j \le m}$. Полето $s_{i,j}$ се означува со точка ако $a_i = b_j$. Непрекинатите дијагонали на точки соодветствуваат на совпаѓања помеѓу секвенците a и b.

На Слика 2.2.1, е прикажана матрицата на точки за пример секвенците $a =$ ACTGTT и $b =$ ACTTT. Дијагоналите: $s_{1,1}s_{2,2}s_{3,3}$, $s_{3,5}s_{4,6}$ и $s_{4,5}s_{5,6}$ ги претставуваат совпаѓањата: ACT и TT.

	b	1	2	3	4	5
a		A	C	T	T	T
1	A	*				
2	C		*			
3	T			*	*	*
4	G					
5	T			*	*	*
6	T			*	*	*

Слика 2.2.1. Матрица на точки

Најдолгото совпаѓање помеѓу две генетски секвенци соодветствува на најдобро сочуван регион. Проблемот на утврдување на најдолго совпаѓање помеѓу две генетски секвенци се решава со примена на *динамичко програмирање*, преку конструкција на матрица $[s_{i,j}]_{1\le i\le n+1, 1\le j\le m+1}$, каде $s_{i,j} = s_{i-1,j-1} + 1$ ако $a_i = b_j$, 0 во спротивност. Полињата $s_{0,j}$ и $s_{i,0}$ се поставени на 0. Најдолгата дијагонала од растечки целобројни вредности, соодветствува на најдолго совпаѓање. Последната вредност долж дијагоналата е должина на совпаѓање.

За пример претходните секвенци, најдолгото совпаѓање е определено со дијагоналата $s_{1,1}s_{2,2}s_{3,3}$: 1 2 3. Последното поле долж претходната дијагонала е полето $s_{3,3} = 3$, што претставува должина на најдолго совпаѓање.

	b	A	C	T	T	T
a	0	0	0	0	0	0
A	0	1	0	0	0	0
C	0	0	2	0	0	0
T	0	0	0	3	1	1
G	0	0	0	0	0	0
T	0	0	0	1	1	1
T	0	0	0	1	2	2

Слика 2.2.2. Утврдување на најдолго совпаѓање

2.3. Краток преглед на алгоритмите за порамнување на генетски секвенци

Еволуциската релација помеѓу две или повеќе генетски секвенци се утврдува со порамнување. Резултатот ги прикажува: *базите на совпаѓање*, *положбите на базна субституција* и *положбите на бришења и додавања на нуклеотиди*. Целта е да се најде порамнување за кое резултатот на порамнување е оптимален. Од друга страна пак, резултатот на порамнување е функција од *награди* и *казни*. Награда се доделува за порамнување на еквивалентни нуклеотиди, додека казна се доделува за порамнување на различни нуклеотиди и додавања на празнини. Вообичаено, казната за порамнување на нуклеотид со празнина се избира да биде повисока од казната што се доделува за порамнување на различни нуклеотиди.

Алгоритмите за порамнување, кои секогаш генерират оптимални решенија, имат неповолна временска и просторна комплексност, што во основа ја ограничува нивната примена само на релативно каратки генетски секвенци. Такви алгоритми се алгоритамот на *Needleman* и *Wunsch* за глобално порамнување и алгоритмот на *Smith* и *Waterman* за локално порамнување на две генетски секвенци. Наспроти претходните алгоритми, современата наука изобилува со хевристички алгоритми, кои генерират резултат за краток временски интервал, со минимална мемориска побарувачка, но добиеното решение не секогаш соодветствува со оптималното. Пример хевристички имплементации се: *BLAST*, *FASTA* и *SPA* (*Super Pairwise Alignment*).

Порамнувањето може да биде: *парово* или *повеќекратно*, *локално* или *глобално*, *празнинско* или *безпразнинско*. Кога се порамнуваат две генетски секвенци порамнувањето е парово, во спротивност повеќекратно. При порамнување на секвенците од крај до крај, порамнувањето е глобално, во спротивност локално. Доколку порамнувањето вклучува додавања и бришења на нуклеотиди порамнувањето е празнинско, во спротвност безпразнинско.

2.4. Алгоритам на Needleman и Wunsch

Првиот алгоритам за порамнување на генетски секвенци е алгоритмот на Needleman и Wunsch за глобално порамнување на две генетски секвенци. Алгоритмот е базиран на динамичко програмирање. Истиот користи матрица на динамичко програмирање $[s_{i,j}]_{0\le i\le n, 0\le j\le m}$, каде n и m се должините на секвенците кои се порамнуваат, $a=\{a_i\}_{i=1}^{n}$ и $b=\{b_j\}_{j=1}^{m}$. Пред да се започне со пресметување на полињата од матрицата на динамичко програмирање, мора да се усвои метрика на порамнување, односно да се определи:

- $s(a_i,b_j)\in Z^+$: награда што се доделува за порамнување на идентични бази, $a_i=b_j$.
- $s(a_i,b_j)\in Z^-$: казна што се доделува за порамнување на различни бази, $a_i\neq b_j$.
- $p\in Z^-$: казна за порамнување на база со празнина, $p<s(a_i,b_j)$.

Полињата $s_{i,0}=i\times p$ и $s_{0,j}=j\times p$ се поставуваат на почеток, за да може да се пресмета остатокот од матрицата на динамичко програмирање. Секое поле од матрицата на динамичко порограмирање со исклучок на полињата од првата редица и колона се пресметува според (2.4.1).

$$s_{i,j}=\max\begin{cases}s_{i-1,j}+p\\ s_{i,j-1}+p\\ s_{i-1,j-1}+s(a_i,b_j)\end{cases} \qquad (2.4.1)$$

Ако $s_{i,j}=s_{i-1,j}+p$, полињата $s_{i-1,j}$ и $s_{i,j}$ се означуваат со вертикален покажувач. Ако $s_{i,j}=s_{i,j-1}+p$, полињата $s_{i,j}$ и $s_{i,j-1}$ се означуваат со хоризонтален покажувач. Ако $s_{i,j}=s_{i-1,j-1}+s(a_i,b_j)$, полињата $s_{i,j}$ и $s_{i-1,j-1}$ се означуваат со дијагонален покажувач. Патеката на покажувачи од полето $s_{n,m}$ до полето $s_{0,0}$ го определува порамнувањето со оптимален резултат на порамнување. Дијагонален покажувач соодветствува на базно порамнување, вертикален покажувач соодветствува на додавање на празнина во секвенцата b, додека хоризонтален покажувач соодветствува на додавање на празнина во секвенцата a.

За да се порамнат пример секвенците: $a = \text{AAAGT}$ и $b = \text{AGT}$ со примена на алгоритмот на Needleman и Wunsch се конструира матрица на динамичко програмирање $[s_{i,j}]_{0\le i\le 5, 0\le j\le 3}$. Како метрика на порамнување се избира $p = -2,\ s(a_i, b_j) = +2$ ако $a_i = b_j$, $s(a_i, b_j) = -1$ ако $a_i \neq b_j$. Полето $s_{1,1}$ се пресметува на начин: $s_{1,1} = \max\{s_{0,1} + p, s_{1,0} + p, s_{0,0} + s(a_1, b_1)\} = \max\{-2-2, -2-2, 0+2\} = \max\{-4, -4, 2\} = 2$. Бидејќи $s_{1,1} = s_{0,0} + s(a_1, b_1) = 3$, полињата $s_{0,0}$ и $s_{1,1}$ се поврзуваат со дијагонален покажувач. Вредноста на полето $s_{1,2}$ изнесува 0. Истата е пресметана на начин $s_{1,2} = \max\{s_{0,2} + p, s_{1,1} + p, s_{0,1} + s(a_1, b_2)\} = \max\{-4-2, 2-2, -2-1\} = 0$. Бидејќи $s_{1,2} = s_{1,1} + p$, полињата $s_{1,2}$ и $s_{1,1}$ се поврзуваат со хоризонтален покажувач. На ист начин се пресметува и остатокот од матрицата на динамичко програмирање. Патеката на испрекинати покажувачи од најдолното десно поле до најгорното лево поле го определува оптималното порамнување. Вертикалните покажувачи соодветствуваат на две последователни додавања на празнини во рамки на секвенцата b по нуклеотидот аденин. Дијагоналните покажувачи соодвтествуваат на совпаѓањата: A и GT. Глобалното порамнување со оптимален резултат е дадено на Слика 2.4.2.

			1	2	3
		b	A	G	T
	a	0	-2	-4	-6
1	A	-2	2	0	-2
2	A	-4	0	1	-1
3	A	-6	-2	-1	0
4	G	-8	-4	0	-2
5	T	-10	-6	-2	2

Слика 2.4.1. Needleman-Wunsch матрица на динамичко порграмирање

```
AAAGT
|  ||
A__GT
```

Слика 2.4.2. Оптимално глобално порамнување за пример секвенците: a и b

2.5. Алгоритам на Smith и Waterman

Алгоритмот на Smith и Waterman генерира оптимално локално порамнување. Решението порамнува фргменти од генетски секвенци, отфрлувајќи ги единечните порамнувања кои го намалуваат резултатот на порамнување. Повторно се конструира матрица на динамичко програмирање $[s_{i,j}]_{0\le i\le n, 0\le j\le m}$, каде полињата се пресметуваат согласно (2.5.1). Вклучувањето на вредноста 0, спречува постоење на полиња со негативна вредност во рамки на матрицата. За разлика од претходно, полињата од првата редица и првата колона се поставени на 0. Порамнувањето се чита од патеката на покажувачи од полето со максимална вредност се до првото нулто поле долж патеката.

За пример претходните секвенци матрицата на динамичко порграмирање е прикажана на Слика 2.5.1. Употребувајќи ја претходно дефинираната метрика на додела на награди и казни, полињата се пресметуваат согласно (2.5.1). На пример, полето $s_{1,1}$ се пресметува како $s_{1,1} = \max\{s_{1,0} + p, s_{0,1} + p, s_{0,0} + s(a_1, b_1), 0\} = \max\{0-2, 0-2, 0+2, 0\} = 2$. Бидејќи $s_{1,1} = s_{0,0} + s(a_1, b_1)$ полињата $s_{0,0}$ и $s_{1,1}$ се поврзуваат со дијагонален покажувач.

Решението го определува патеката на покажувачи од полето со максимална вредност се до првото нулто поле. Полето со максимална вредност претставува вредност на оптимално локално порамнување. Трите дијагонални покажувачи долж патеката соодветствуваат на три базни порамнувања на последните три нуклеотиди. Порамнувањето е дадено на Слика 2.5.2.

$$s_{i,j} = \max\begin{cases} s_{i-1,j} + p \\ s_{i,j-1} + p \\ s_{i-1,j-1} + s(a_i, b_j) \\ 0 \end{cases} \quad (2.5.1)$$

			1	2	3
		b	A	G	T
	a	0	0	0	0
1	A	0	2	0	0
2	A	0	2	1	0
3	A	0	2	1	0
4	G	0	0	4	0
5	T	0	0	2	6

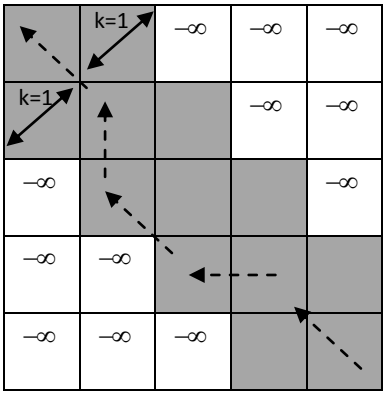
Слика 2.5.1. Smith-Waterman матрица на динамичко програмирање

```
A G T
| | |
A G T
```

Слика 2.5.2. Оптимално глобално порамнување за пример секвенците: a и b

2.6. Порамнување во дијагонално симетричен опсег

При порамнување на генетски секвенци со висок процент на базна идентичност, решението е определено со патека на покажувачи која конвергира околу главната дијагонала од матрица на динамичко програмирање. Ако оптималното порамнување е определено со патека на покажувачи, која се оддалечува за не повеќе од k полиња, релативно во однос на главата дијагонала, тогаш наместо mxn полиња се пресметуваат kxn (k<m) полиња, на растојание помало или еднакво на k , во однос на главната дијагонала, (1.3), Слика 2.6.1. Полињата на растојание поголемо од k се поставени на $-\infty$ и истите ниту се пресметуваат, ниту пак се чуват во меморија, што ја намалува временската и просторната комплексност на порамнување од O(mn) на O(kn), каде k е должината на дијагонално симетричниот опсег, каде се бара решението.



Слика 2.6.1. Порамнување во дијагонално симетричен опсег

$$s_{i,j} = \begin{cases} s_{i,j}, |\, i-j \,| \le k \\ -\infty, |\, i-j \,| > k \end{cases} \quad (1.3)$$

Должината на дијагонално симетричниот опсег е непозната. Земајќи минимална, позитивна и целобројна вредност за k, оптимално решенение во рамки на дијагонално симтеричниот опсег е воедно и глобално оптимално решение ако и само ако е задоволено неравенството: $s > 2(k+1)p + (n-(k+1))\alpha$, каде s е резултатот на оптимално порамнување во рамки на дијагонално симетричен опсег со должина k, p е казна за порамнување на база со празнина и α е резултат на единечно базно порамнување. Претходното неравенство важи за случајот кога се порамнуваат генетски секвенци со еднаква должина.

Употребувајќи ја претходната метрика на порамнување, пример секвенците a: AACA и b: AAAT се порамнуваат глобално во рамки на дијагонално симетричен опсег, со почетен избор за k=1. Се пресметуваат само полињата $s_{i,j}, |\, i-j \,| \le 1$. За полињата $s_{i,j}, |\, i-j \,| > 1$ се зема дека се поставени на $-\infty$ и истите немат влијание врз пресметките на вредностите за полињата $s_{i,j}, |\, i-j \,| \le 1$, бидејќи секогаш се избира максимумот од три вредности: $s_{i-1,j} + p, s_{i,j-1} + p, s_{i-1,j-1} + s(a_i, b_j)$, што во никој случај не може да биде $-\infty$.

Постојат две различни глобални порамнувања со резултат на порамнување 2. Бидејќи важи $s = 2 > -4 = 2(k+1)p + (n-(k+1))\alpha$, добиените решенија се воедно и глобално-оптимални решенија, односно не постои порамнување определено со патека на покажувачи, која излегува надвор од дијагонално симетричниот опсег со должина 1, со поповолен резултат на порамнување од 2.

	b	A	A	A	T
a	0	-2			
A	-2	2	0		
A		0	4	2	
C			2	3	1
A				4	2

Слика 2.6.1. Порамнување во дијагонално симетричен опсег за пример секвенците: a и b

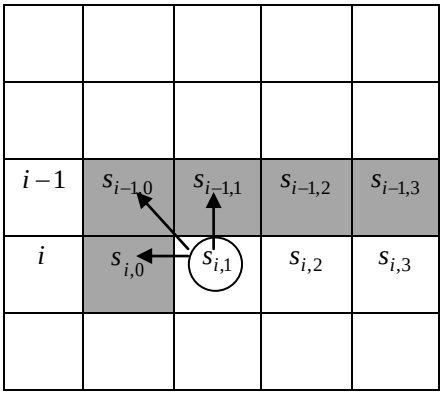
```
AACA      AACA_
||        || |
AAAT      AA_AT
```

Слика 2.6.2. Приказ на решенија

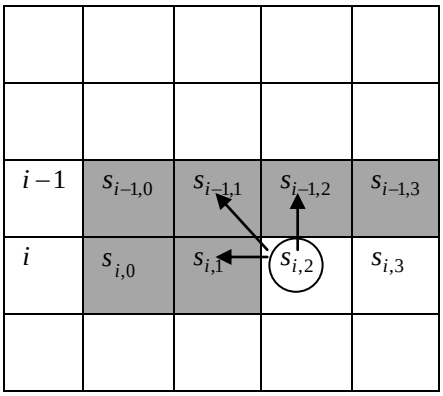
2.7. Пресметка на оптимален резултат на порамнување со линеарен мемориски трошок

Вредноста на поле $s_{i,j}$ од матрица на динамичко програмирање зависи од вредностите на полињата: $s_{i-1,j}, s_{i,j-1}$ и $s_{i-1,j-1}$. Полето $s_{i,j-1}$ се наоѓа во иста редица со полето $s_{i,j}$, додека останатите полиња $s_{i-1,j}$ и $s_{i-1,j-1}$ се наоѓаат во претходната редица. За да се пресмета полето $s_{i,j}$ доволно е да се чуваат во меморија редиците $i-1$ и i, бидејќи вредностите од останатите редици немат удел при пресметката на вредноста на полето $s_{i,j}$.

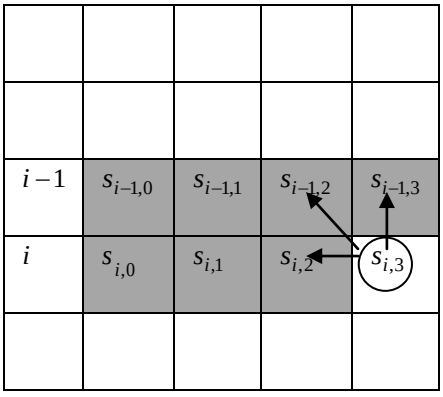
Чувајќи две по две редици во меморија во фазата на извршување, вредноста на оптимално порамнување, определена со полето $s_{n,m}$, се пресметува со линеарен мемориски трошок. Бидејќи во меморија се чуваат две по две редици, односно при пресметување на вредностите од редица i учествуваат вредностите од претходната редица $i-1$ и претходно пресметаните вредности од редицата i, извршувајќи ги пресметките од лево па на десно, се губи патеката на покажувачи која го определува оптималното порамнување. Тоа значи дека со линеарен мемориски трошок може да се пресмета вредноста на оптимално порамнување, но не и оптималното порамнување, Слика 2.7.1, Слика 2.7.2 и Слика 2.7.3.



Слика 2.7.1. Пресметка на вредностите од редица i



Слика 2.7.2. Пресметка на вредностите од редица i



Слика 2.7.3. Пресметка на вредностите од редица i

Употребувајќи ја претходно воведената метрика, вредноста на оптимално порамнување за пример секвенците a: ACT и b: ACC може да се пресмета во линеарен простор. Со меморирање на вредностите од нултата редица може да се пресметат

вредностите на полињата од првата редица. Отако ќе се пресметат сите полиња од првата редица, нултата редица се брише од меморија. Вредностите на полињата од втората редица зависат од претходно пресметаните вредности од истата редица и вредностите на полињата од првата редица. По пресметување на вредноста на полето $s_{2,3}$, првата редица се брише од меморија и се оди на пресметување на вредностите од третата редица. Вредноста на полето $s_{3,3} = 3$ е вредност на оптимално порамнување за пример секвенците: a: ACT и b: ACC. Во фаза на извршување во меморијата се чуваат минимум 4 м.е (м.е: мемориска единица=4B), максимум 8 м.е. Доколку во меморија би се чувала комплетната матрица на динамичко програмирање, неопходно е уште на самиот почеток да се резервират 16 мемориски единици.

		0	1	2	3
			A	C	C
0		0	-2	-4	-6
1	A	-2	2	0	-2
2	C				
3	T				

Слика 2.7.4. Пресметување на вредностите на полињата од првата редица

		0	1	2	3
			A	C	C
0					
1	A	-2	2	0	-2
2	C	-4	0	4	2
3	T				

Слика 2.7.5. Пресметување на вредностите на полињата од втората редица

		0	1	2	3
			A	C	C
0					
1	A				
2	C	-4	0	4	2
3	T	-6	-2	2	3

Слика 2.7.6. Пресметување на вредностите на полињата од третата редица

2.8. Алгоритам на Hirschberg

Просторниот трошок може да се намали од O(nm) на O(n+m) со примена на алгоритмот на Hirschberg. Базиран на примена на стратегијата *разделu па владеј*, во секој чекор се бара точка на пресек (x,y), која ги дели секвенците $a = \{a_i\}_{i=1}^{n}$ и $b = \{b_j\}_{j=1}^{m}$ на подсеквенци: $a_l = a_1 a_2 .. a_{x-1} a_x$, $b_l = b_1 b_2 .. b_{y-1} b_y$ и $a_r = a_{x+1} a_{x+1} .. a_{n-1} a_n$, $b_r = b_{y+1} b_{y+1} .. b_{m-1} b_m$. Точката на пресек (x,y) овозможува конструкција на оптимално порамнување за секвенците a и b со соединување на оптималните порамнувања за подсеквенците a_l, b_l и a_r, b_r. Точки на пресек се наоѓаат за секој пар на леви и десни подсеквенци, се до единечено базно порамнување или порамнување на база со празнина. Со враќање назад се добива оптималното порамнување.

За да се најде точката на пресек (x,y) се пресметуваат последните две редици $r_{1,f}$ и $r_{2,f}$ од матриците на динамичко програмирање за паровите секвенци: $a_1 .. a_{[\frac{n}{2}]}, b_1 .. b_m$ и $a_n .. a_{[\frac{n}{2}]+1}, b_m .. b_1$. Пресметките се извршуваат со меморирање на две по две редици, со што се линеаризира просторниот трошок. Индексот на елементот со максимална вредност од векторот $v = r_{1,f} + r_{2,f}^{reverse}$, каде $r_{2,f}^{reverse}$ е редицата $r_{2,f}$ испишана во обратен редослед, го определува пресекот по колона y. На тој начин се добива точка на пресек $(x, y) = (\frac{n}{2}, index\ of\ \max(r_{1,f} + r_{2,f}^{reverse}))$.

Врз основа на претходно усвоената метрика на порамнување, пример секвенците a: AACG и b: ACG се порамнуваат со примена на алгоритмот на Hirschberg. Првата точка на пресек е точката $(\left[\frac{n}{2}\right], y) = (\left[\frac{4}{2}\right], y) = (2, y)$. За да се најде индексот на пресекот по колона се

порамнуваат паровите секвенци: AA, ACG и GC, GCA. Последните две редици од матриците на динамичко програмирање се редиците: $r_{1,f} = [-2,0,1,-1]$ и $r_{2,f} = [-2,0,4,2]$, од каде за векторот v се добива: $v = [0,4,1,-3]$ (забележете дека $r_{2,f}^{reverse} = [2,4,0,-2]$). Индексот на елементот со максимална вредност 4 од векторот v е 1, од каде за точката на пресек се добива (x,y)=(2,1), што значи дека оптималното порамнување за секвенците a и b се добива со соединување на оптималните порамнувања за паровите подсеквенци: AA,A и CG,CG.

		A	C	G
	0	-2	-4	-6
A	-2	2	0	-2
A	-2	0	1	-1

Слика 2.8.1. Делење на матрица на динамичко програмирање

		G	C	A
	0	-2	-4	-6
G	-2	2	0	-2
C	-2	0	4	2

Слика 2.8.2. Делење на матрица на динамичко програмирање

Табела 2.8.1. Пресметување на индекс на колона на делење

Индекс:	0	**1**	2	3
$r_{1,f}$	-2	0	1	-1
$r_{2,f}^{reverse}$	2	4	0	-2
$\sum$	0	4	1	-3

На ист начин се наоѓаат точки на пресек за десниот пар подсеквенци: CG,CG и левиот пар подсеквенци: AA, A. Точката на пресек за десниот пар подсекевцни CG,CG е точката (1,1), што значи дека оптималното порамнување за секвенците CG,CG се добива со соединување на

единечните порамнувања C:C и G:G. Точката на пресек за вториот пар подсеквенци AA,A е точката (1,0), што значи дека вториот аденин од подсеквенцата AA се порамнува со аденин подсеквенцата A, додека првиот се порамнува со празнина. Делењето овде запира. Оптималното порамнување за секвенците a и b се добива со соединување на под-проблемските решенија, Слика 2.8.7.

		C	G
	0	-2	-4
C	-2	2	0

Слика 2.8.3. Делење на матрицата на динамичко програмирање за подсеквенци

		G	C
	0	-2	-4
G	-2	2	0

Слика 2.8.4. Делење на матрицата на динамичко програмирање за подсеквенци

Табела 2.8.2. Пресметување на индекс на колона на делење

Индекс:	0	**1**	2
$r_{1,f}$	-2	2	0
$r_{2,f}^{reverse}$	0	2	-2
$\sum$	-2	4	-2

		A
	0	-2
A	-2	2

Слика 2.8.5. Делење на матрицата на динамичко програмирање за подсеквенци

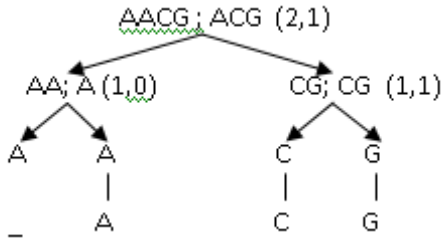
		A

	0	-2
A	-2	2

Слика 2.8.6. Делење на матрицата на динамичко програмирање за подсеквенци

Табела 2.8.3. Пресметување на индекс на колона на делење

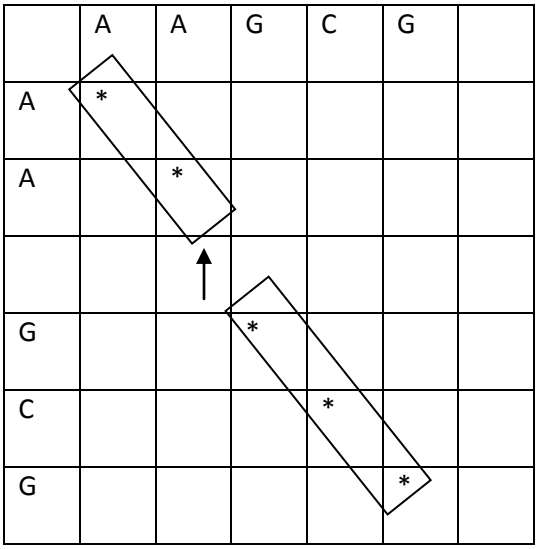
Индекс:		0	**1**
$r_{1,f}$		-2	2
$r_{2,f}^{reverse}$		2	-2
$\sum$		0	0



Слика 2.8.7. Соединување на подпроблемските решенија

2.9. FASTA

Совпаѓањата помеѓу две генетски секвенци учествуваат во градбата на локалното порамнување. Фаста наоѓа совпаѓања помеѓу прашалник секвенца q и целна секвенца t од генетска база на податоци. Совпаѓањата соодветствуваат на непрекинати дијагонали од матрица на точки, Слика 2.9.1. Со соединување на дијагоналите се образува локално порамнување.



Слика 2.9.1. Идентификација на совпаѓања преку матрица на точки

Постапката започнува со издвојување на преклопувачки зборови со должина l од прашалник секвенца q. Се барат совпаѓања на издвоените зборови во рамки на целната секвенца. Почетната позиција на секое збор совпаѓање, како во прашалник секвенцата, така и во целната секвенца се бележи во табела. Пресметувајќи ги разликите помеѓу почетните положби на збор совпаѓањата во рамки на прашалник и целната секвенца, се барат разликите со највисока честота. Разликите од соседните редици со висока честота соодветствуваат на значајни совпаѓања. Со соединување на значајните совпаѓања се образува локалното порамнување. Соединувањето е овозможено со додавање на една или повеќе празнини.

Примената на Фаста ќе ја покажеме за пример прашалник секвенцата q: ACAGCGCT и пример целната секвенца t: GACAG**C**TTA. Ако за должина на зборовите се земе $l = 2$, од прашалник секвенцата може да се издвојат $|q| - l + 1 = 8 - 2 + 1 = 7$ преклопувачки зборови: AC, CA, AG, GC, CG, GC и CT. Се барат совпаѓања на овие зборови во рамки на целната секвенца, бележејќи ја почетната положба на секое совпаѓање во рамки на прашалник и целната секвенца, Табела 2.9.1.

Табела 2.9.1. Почетни положби на заедничките совпаѓања

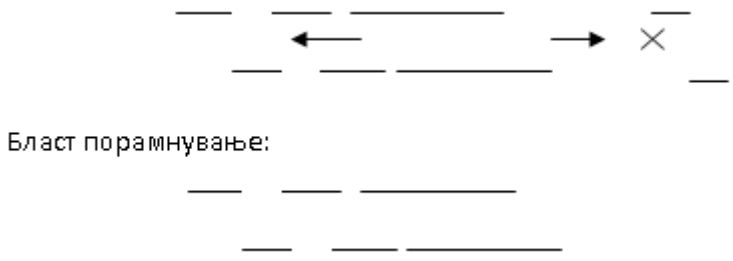
Збор	Положба во q	Положба во t	Разлика
AC	1	2	1-2=-**1**
CA	2	3	2-3=-**1**
AG	3	4	3-4=-**1**
GC	4, 6	5	4-5=-**1**, 6-5=1
CG	5	/	
CT	7	6	7-6=1

Разликата -1 е е разлика со највисока честота. Истата последователно се повторува четири пати, што соодветствува на совпаѓање со должина од пет нуклеотиди: ACAGC помеѓу прашалник и целната секвенца. Совпаѓањето ACAGC е Фаста локално порамнување, Слика 2.9.1.

Слика 2.9.1. Фаста локално порамнување

2.10. БЛАСТ

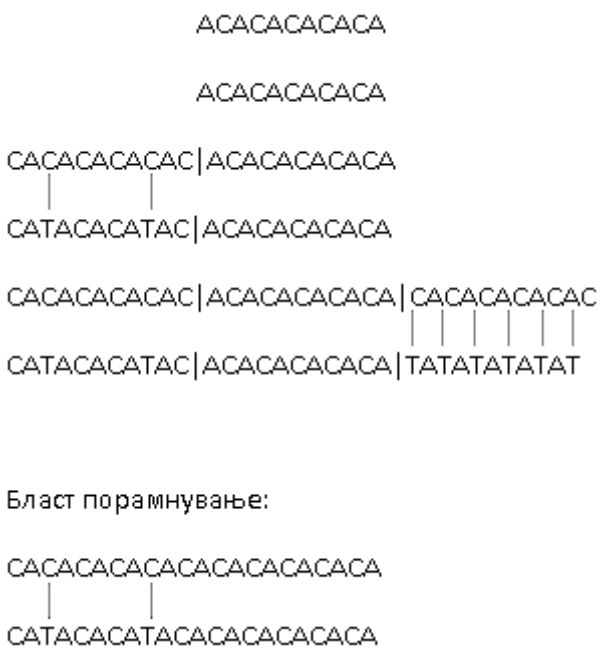
Бласт пребарува генетска база на податоци по прашалник секвенца. Резултатот од пребарувањето е листа на секвенци со највисок процент на сличност во однос на прашалник секвенца. Безпразнинскиот Бласт работи на принцип на идентификација на 11-базно почетно совпаѓање помеѓу прашалник и целна секвенца од база на податоци. Почетното 11-базно совпаѓање се проширува на лево и на десно за 11-базни зборови, се додека резултатот на порамнување се зголемува. Усвојувајќи метрика на порамнување која доделува награда $+1$ за вклучување на еквивалентно порамнети бази и казна -1 за вклучување на нееквиваленто порамнети бази, резултатот на порамнување ќе расте се додека бројот на додадени еквивалентно порамнети бази е поголем од бројот на додадени нееквивалентно порамнети бази, Слика 2.10.1.



Слика 2.10.1. БЛАСТ порамнување

Примената на Бласт ќе ја покажеме за пример прашалник секвенцата q: CACACACACAC|ACACACACACA|CACACACACAC и целна секвенца t: TGT|CATACACATAC|ACACACACACA|TATATATATATGT. Совпаѓањето ACACACACACA е 11-базно совпаѓање помеѓу прашалник и целната секвенца, од каде започнува издолжувањето. Употребувајќи метрика на порамнување која доделува награда +1 за еквивалентно порамнети бази и казна -1 за нееквивалентно порамнети бази, почетната вредност на порамнувањето,

определено со ACACACACACA совпаѓањето изнесува 11. Со издолжување за 11 базни позиции на лево се образува порамнување со резултат на порамнување 20=11+9, бидејќи од новододадените единаесет базни порамнувања девет базни порамнувања се еквивалентни. Издолжувајќи на десно за 11 базни позиции, резултатот на порамнување се намалува од 20 на 19, бидејќи бројот на додадени нееквивалентно порамнети базни парови е поголем од бројот на додадени еквивалентно проаменти базни парови. Поради тоа последното 11-базно издолжување се отфрла. Понатамошно базно издолжување не е можно, што резултира со решение на Слика 2.10.2.



Слика 2.10.2. Пример БЛАСТ порамнување

2.11. Алгоритми за повеќекратно порамнување: *КлусталВ* и *Ѕвезда порамнување*

КлусталВ и ѕвезда порамнувањето се пристапи за порамнување на повеќе генетски секвенци. Се бара порамнување со оптимален резултат на порамнување за повеќе од две секвенци. Ниту една колона од резултатот не смее да порамнува исклучиво празнини. Метриката: *збир на сума на парови по секоја колона* се употребува за да се пресмета резултатот на порамнување. Задржувајќи ја првично воведената метрика на додела на награди и казни, сумата на парови за првата колоната за пример порамнувањето на Слика 2.11.1 изнесува -2. Истата се пресметува на начин: s_1=s(A,_)+s(A,A)+s(_,A)=-2+2-2=-2. Аденинот од првата секвенца се порамнува со празнина од втората секвенца и аденин од третата секвенца, додека празнината од втората секвенца се порамнува со аденин од третата секвенца. На идентичен начин може да се пресмета и сумата на парови по втората колона

s_2=s(G,G)+s(G,G)+s(G,G)=2+2+2=6 и третата колона s_3=s(_,_)+s(_,T)+s(_,_)=0+0+-2=-2. Додела на вредност 0 за празнинско парово порамнување елиминира додатна додела на казна при додатно вклучување на празнини. Резултатот на порамнување се пресметува како збир на сумите на парови по трите колони, s=s_1+s_2+s_3=-2+6-2=2.

AGT_

_GC_

AGTT

Слика 2.11.1. Пример за повеќекратно порамнување

КлусталВ (ClustalW) е пристап за порамнување на повеќе од две генетски секвенци. Со примена на алгоритмот на *Needleman* и *Wunsch* се порамнува секој пар секвенци $S_i, S_j, 1 \le i, j \le n, i \ne j$ од множеството секвенци $S_1, S_2, ..., S_{n-1} S_n$. За секое порамнување S_i, S_j се пресметува растојание $d_{i,j}$ како количник помеѓу бројот на порамнети базни несовпаѓања и вкупниот број на базни порамнувања, не броејќи ги порамнувањата база со празнина. Растојанијата $d_{i,j}$ се додаваат во рамки на симетрична матрица на растојанија $[d_{i,j}]_{1 \le i,j \le n}$ $d_{i,j} = d_{j,i}$. Врз основа на податоците од матрицата на растојанија се гради водечко стебло, кое го одредува редоследот на комбинирање на паровите порамнувања во фазата на констукција на повеќекратно порамнување.

Примената на КлусталВ ќе ја покажеме на пример секвенците: S_1 : ACCG, S_2 : ACG, S_3 : CCT и S_4 : CGT. Паровите секвенци: $S_1, S_2; S_1 S_3; S_1 S_4; S_2 S_3; S_2 S_4; S_3 S_4$ се порамнуваат со примена на алгоритмот на *Needleman* и *Wunsch*, употребувајќи ја почетно воведената метрика на порамнување. Оптималните порамнувања и растојанијата помеѓу паровите секвенци се дадени во продолжение.

S_1 : ACCG

S_2 : AC_G $d_{1,2} = \frac{0}{3}$

S_1 : ACCG

S_3 : _CCT $d_{1,3} = \frac{1}{3}$

S_1 : ACCG

S_4 : _CGT $d_{1,4} = \frac{2}{3}$

S_2 : ACG

S_3 : CCT $d_{2,3} = \frac{2}{3}$

S_2 : ACG_

S_4 : _CGT $d_{2,4} = \frac{0}{2} = 0$

S_3 : CCT

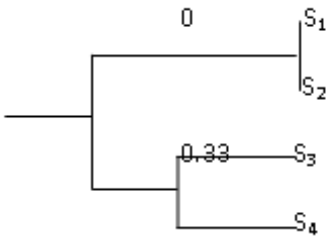
S_4 : CGT $d_{3,4} = \frac{1}{3}$

Растојанието помеѓу секвенците S_1 : ACCG и S_2 : ACG изнесува $d_{1,2} = \frac{0}{3} = 0$ (не постои базно порамнување помеѓу различни нуклеотиди). Растојанието помеѓу секвенците S_1 : ACCG и S_3 : CCT изнесува $d_{1,3} = \frac{1}{3}$ (едно нееквивалентно базно порамнување од три базни порамнувања).

	S_1	S_2	S_3	S_4
S_1				
S_2	0			
S_3	1/3	2/3		
S_4	2/3	0	1/3	

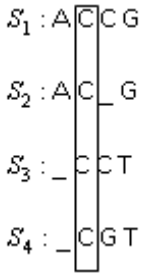
Слика 2.11.2. Матрица на растојанија

Врз основа на податоците од матрицата на растојанија се образува водечко стебло – Слика 2.11.3, кое го определува редоследот на комбинирање на паровите порамнувања во фазата на образување на повеќекратно порамнување.



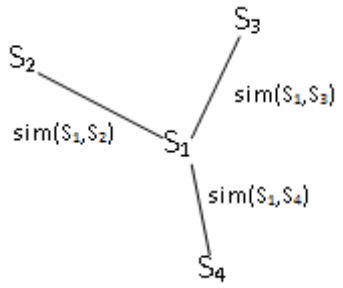
Слика 2.11.3. Водечко стебло

Бидејќи растојанието помеѓу секвенците S_1 и S_2 е најмало, S_1;S_2 порамнувањето се вклучува на почеток. Должината на порамнување изнесува 4. Ова порамнување се комбинира со паровото порамнување S_3; S_4 , со должина 3. Не менувајќи ја структурата на S_3;S_4 порамнувањето, за да се образува повеќекратно порамнување со должина 4, на почеток или на крај од S_3 и S_4 треба да се додаде по една празнина. Порамнување со подобар резултат на порамнување се образува ако истите се додадат на почеток.



Слика 2.11.4. КлусталВ повеќекратно порамнување

Ѕвезда пристапот за повеќекратно порамнување бара секвенца S_i со најголем степен на сличност со една од останатите секвенци $S_j, i \neq j$. За степен на сличност помеѓу секвенците S_i и S_j се избира резултатот на оптимално парово порамнување. За ѕвезда секвенца се избира секвенца S_i за која збирот (2.11.1) е максимален, каде $sim(S_i, S_j)$ е резултатот на оптимално парово порамнување за секвенците S_i и S_j . По определување на ѕвезда секвенцата, повеќекратното порамнување се образува со додавање на паровите порамнувања помеѓу ѕвезда секвенцата S_i и остатокот секвенци. Во фазата на образување на повеќекратно порамнување се врши и усогласување помеѓу порамнувањата со различна должина. Усогласувањето се врши со додавање на празнини на почеток или на крај од секвенците.



Слика 2.11.4. Избор на ѕвезда секвенца

$$Z_i = \sum_{j=1, j\neq i}^{n} sim(S_i, S_j) \qquad (2.11.1)$$

Се разгледува ѕвезда порамнување за пример секвенците: S_1 : ACCG, S_2 : ACG, S_3 : CCT и S_4 : CGT од претходниот пример, употребувајќи ја почетната метрика на порамнување. За секој пар S_i, S_j се пресметува сличноста $sim(S_i, S_j)$ како резултат на оптимално порамнување за секвенците S_i и S_j. Шестте комбинации на парови порамнувања и сличностите $sim(S_i, S_j)$ се дадени во продолжение. За четирите пример секвенци се пресметуваат збировите Z_1, Z_2, Z_3 и Z_4, при што за ѕвезда секвенца се избира секвенцата S_i за која важи дека збирот Z_i е максимален. Овде за ѕвезда секвенца може да се избере S_2 или S_3. Во случај на избор на секвенцата S_2, повеќекратното порамнување се образува со додавање на порамнетите секвенци S_1, S_4 и S_3 во однос на секвенцата S_2. При последното додавање се врши усогласување помеѓу должините на порамнување со додавање на една празнина на почеток на S_3.

S_1 : ACCG

S_2 : AC_G $sim(S_1, S_2) = 2 + 2 - 2 + 2 = 6 - 2 = 4$

S_1 : ACCG

S_3 : _CCT $sim(S_1, S_3) = -2 + 2 + 2 - 1 = 4 - 3 = 1$

S_1 : ACCG

S_4 : _CGT $sim(S_1, S_4) = -2 + 2 - 1 - 1 = -2$

S_2 : ACG

S_3 : CCT $sim(S_2, S_3) = -1 + 2 - 1 = 0$

S_2 : ACG_

S_4 : _CGT $sim(S_2, S_4) = -2 + 2 + 2 - 2 = 0$

S_3 : CCT

S_4 : CGT $sim(S_3, S_4) = 2 - 1 + 2 = 3$

$Z_1 = sim(S_1, S_2) + sim(S_1, S_3) + sim(S_1, S_4) = 4 + 1 - 2 = 3$

$Z_2 = sim(S_2, S_1) + sim(S_2, S_3) + sim(S_2, S_4) = 4 + 0 + 0 = 4$

$Z_3 = sim(S_3, S_1) + sim(S_3, S_2) + sim(S_3, S_4) = 1 + 0 + 3 = 4$

$Z_4 = sim(S_4, S_1) + sim(S_4, S_2) + sim(S_4, S_3) = -2 + 0 + 3 = 1$

S_1 : ACCG

S_2 : AC_G

S_1 : ACCG

S_2 : AC_G

S_4 : _CGT

S_1 : ACCG

S_2 : AC_G

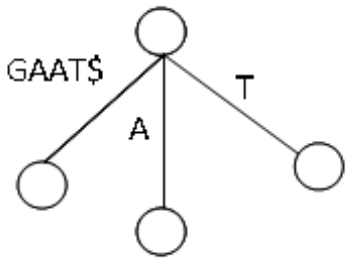
S_4 : _CGT

S_3 : _CCT

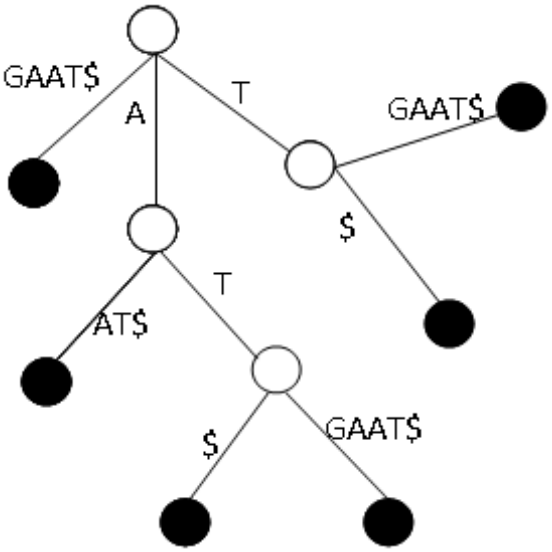
Слика 2.11.5. Повеќекратно порамнување според Ѕвезда пристапот

3. Пребарување на генетски секвенци и генетски бази на податоци, *стебла на суфикси* и *SSAHA пребарување*

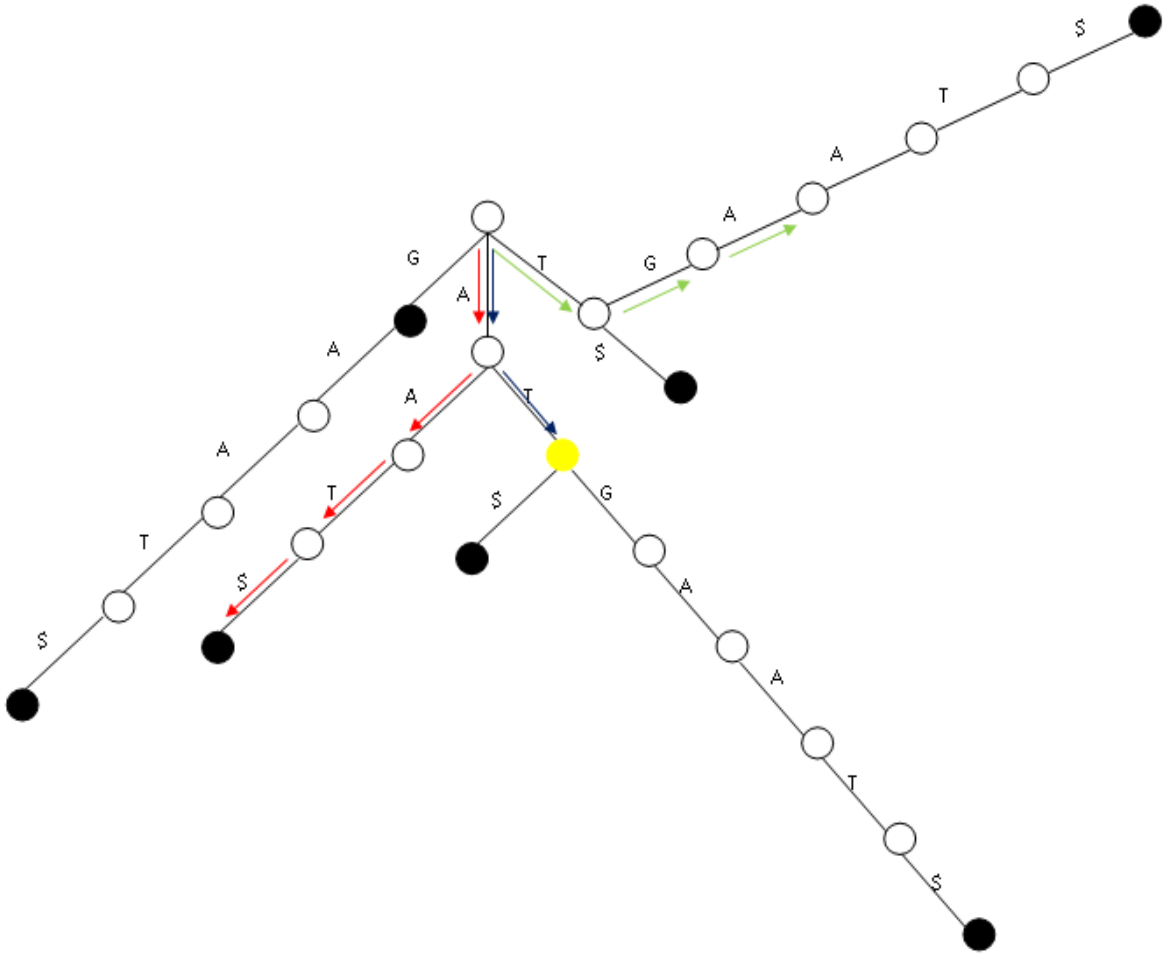
Некои биоинформатички проблеми се решаваат со употреба на стебло на суфикси. Стебло на суфикси за секвенца со должина n се конструира во O(n) време и зафаќа O(n) меморија. За да се изгради стебло на суфикси за секвенца со должина n се издвојуваат исто толку различни суфикс секвенци. За пример секвенцата ATGAAT можат да се издвојат 6 суфикс секвенци: ATGAAT, TGAAT, GAAT, AAT, AT, T. Во зависност од почетниот карактер, три суфикс секвенци започнуваат со карактерот A: ATGAAT,AAT и AT, две со T: TGAAT и T и една со G: GAAT, врз основа на што се определува почетното разгранување, Слика 3.1. Карактерот на позиција 2 кај суфикс секвенците што започнуваат со A е A или T, што резултира со двојно разгранување на јазелот поврзан со A-означената гранка на AT$ и T гранки (знакот $ означува крај на суфикс секвенца), Слика 3.2. Нетерминалниот јазел T понатаму се разгранува на $ и GAAT$, Слика 3.2. По аналогија на претходното нетерминалниот јазел T од првото ниво на разгранување се грани на $ и GAAT$, Слика 3.2.



Слика 3.1. Почетно разгранување на стебло на суфикси



Слика 3.2. Стебло на суфикси за пример секвенцата: ATGAAT



Слика 3.3. Стебло на суфикси за пример секвенцата: ATGAAT

По конструкција на стебло на суфикси за секвенца S, со изминување може да се утврди: ***дали секвенца*** q ***е подстринг или суфикс во*** S, ***да се најде бројот на повторувања на*** q ***во*** S и ***да се утврди најдолгото повторување***.

За да се провери дали секвенца q е подстринг во S се следи патеката на q започнувајќи од коренот на стеблото. Ако патеката ја покрива во целост содржината на прашалник секвенцата q, тогаш q е подсеквенца од S. Ако за претходната пример секвенца S: ATGAAT се провери дали секвенцата q: TGA е подстринг, тогаш одговорот е потврден, бидејќи постои патека TGA во рамки на стеблото на суфикси (на Слика 3.3 означена со патека на зелени покажувачи).

За да се провери дали секвенца q е суфикс во S се следи повторно патеката на q започнувајќи од коренот на стеблото. Ако патеката ја покрива во целост содржината на прашалник секвенцата q и истата завршува со терминален јазел, тогаш q е суфикс во S. За да се провери дали секвенцата q: AAT е суфикс во S се следи патеката на секвенцата q (на Слика 3.3 означена со патека на црвени покажувачи). Бидејќи патеката завршува со терминален јазел (означен со црна боја), одоговорот на прашањето е потврден. За разлика од секвенцата AAT, секвенца TGA не претставува суфикс, бидејќи патеката на истата завршува во нетерминален јазел (означен со бела внатрешна обоеност).

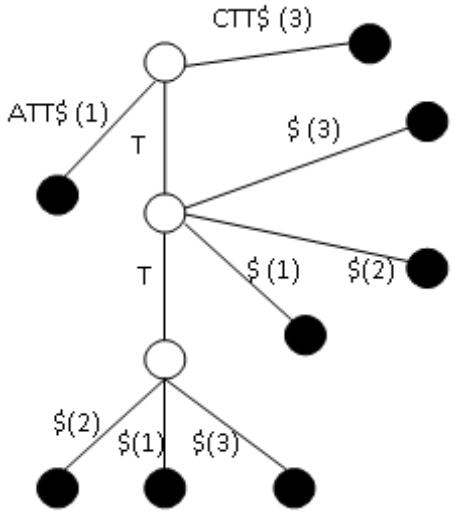
За да се најде бројот на повторувања на q во S повторно се следи патеката на q се до нејзиниот крај. Бројот на јазли под последниот јазел од патеката на прашалник секвенцата q претставува број на повторувања на секевнцата q во S. Прашалник секвенцата q: AT се повторува два пати во S. До овој заклучок може да се дојде следејќи ја AT патеката. Бидејќи бројот на јазли под јазелот T е 2, бројот на повторувања на секвенцата AT во S е исто така 2 (Слика 3.3 - патека на сини покажувачи).

За да се утврди најдолгото повторување во S се бара најдлабок јазел, со најмалку два јазли под него. За пример секвенцата S: ATGAAT, AT повторувањето е најдолго повторување. До претходниот заклучок се доаѓа преку јазелот поврзан со T-означената гранка долж AT патеката, кој е најдлабок јазел (означен со жолта боја на Слика 3.3) во рамки на стеблото на суфикси, со два јазли под него.

Стебло на суфикси може да се конструира за повеќе од една секвенца. Генерализираните стебла на суфикси се погодна податочна структура за репрезентација и пребарување на множество секвенци. Пребарувајќи ја податочната структура може да се утврди најдолгото заедничко совпаѓање помеѓу $k, k \geq 2$ секвенци. Покрај претходниот

проблем, се решава и проблемот на утврдување на содржајност на прашалник секвенца q во рамки на множество од $k, k \geq 2$ секвенци.

За да се конструира стебло на суфикси за $k, k \geq 2$ секвенци, се издвојува и групира секоја суфикс секвенца од множеството на секвенци. При издвојување на суфикс секвенците се вклучува и додатна информација: индекс на секвенца од каде суфикс секвенцата потекнува. На Слика 3.4 е претставено генерализираното стебло на суфикси за секвенците: S_1 : ATT, S_2 : TT и S_3 : CTT.



Слика 3.4. Генерализирано стебло на суфикси за пример секвенците: S_1, S_2 и S_3

Најдолгото заедничко совпаѓање за претходните секвенци е TT совпаѓањето, кое го определува најдлабокиот јазел од каде потекнуваат: $(1), $(2) и $(3)-означени гранки.

За да се утврди припадност на прашалник секвенца q, се бара последниот јазел долж патеката на прашалникот. Секоја $(x) означена гранка, која потекнува од претходниот јазел, определува секвенца x, која ја содржи прашалник секвенца. Прашалник секвенца q : TT е заеднички подстринг за сите три секвенци, односно од последниот јазел долж TT патеката потекнуваат $(1), $(2) и $(3)-означени гранки.

3.1. MUMmer

MUMmer е хевристички пристап за порамнување на долги ДНК секвенци, најчесто комплетни геноми a и b, базиран на употреба на стебло на суфикси. Порамнувањето се врши во три фази:

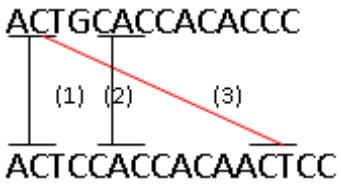
Фаза 1: *Идентификација на единствени совпаѓања со максимална должина*.

Фаза 2: *Селекција на најдолго множество на совпаѓања со максимална должина, кои се појавуваат во ист редослед во двете секвенци, според принципот на утврдување на најдолго растечка подсеквенца*.

Фаза 3: *Затворање на празнини со примена на динамичко програмирање*.

По дефиниција, единствено совпаѓање со максимална должина е подсеквенца која се појавува еднаш во рамки на секвенците a и b и не е дел од друга подолга подсеквенца. Така на пример, секвенцата ACT е единствено совпаѓање со максимална должина за пример секвенците секвенците: a : ACTGCACCACACCC и b : ACTCCACCACAACTCC. Совпаѓањата како: AC и CT не се единствени совпаѓања со макисимална должина, бидејќи истите се дел од подолга совпаѓачка секвенца. CCA совпаѓањето исто така не претставува единствено совпаѓање, бидејќи истото се повторува два пати во рамки на секвенцата b.

MUMmer ги наоѓа единствените совпаѓања со максимална должина преку стебло на суфикси. Од единствените совпаѓања со максимална должина, се издвојува множество на единствени совпѓања, кои се појавуваат во ист редослед во рамки на секвенците: a и b. За пример секвенците, множеството на единствени совпаѓања со максимална должина ги вклучува совпаѓањата (1) и (2), Слика 3.1.1. Совпаѓањето (3), на Слика 3.1.1 означено со црвена боја, не се вклучува во рамки на множеството на единствени совпаѓања, бидејќи истото е во пресек со совпаѓањето (2).



Слика 3.1.1. Единствени совпаѓања со максимална должина

Во третата фаза се затвараат локалните празнини, според алгоритмот на Smith и Waterman, при што се издвојуваат четири категории на празнини:

- ***Единечни нуклеотидини полиморфизм***: единечни базни несовпаѓања помеѓу две последователни единствени совпаѓања од множеството на единствени совпаѓања со максимална должина. За пример секвенците, G-C базното несовпаѓање претставува единечен нуклеотиден полиморфизам, Слика 3.1.3.

- ***Додавања***: подсеквенци кои се појавуваат само во рамки на една од секвенците. За пример секвенците, подсеквенцата ACT на крај од секвенцата b претставува додавање, бидејќи истата се појавува во рамки на b, но не и во a. Додавањата се порамнуваат со додавање на празнини во рамки на спротивната секвенца.

- ***Кратки тандемски повторувања***: заеднички подсеквенци, кои се повторуваат повеќе пати. За пример секвенците, CA повторувањето, претставува кратко тандемско повторување.

- ***Региони на полиморфизам***: картки ДНК фрагменти со висока стапка на мутација. ДНК фрагментот CCC од секвенцата a и CC фрагментот од секвенцата b се региони на полиморифизам и истите се порамнуваат со примена на динамичко програмирање, Слика 3.1.2.

```
CCC
_CC
```

Слика 3.1.2. Региони на полиморфизам

По затворањето на четирите категории на празнини, се печати MUMmer порамнувањето, Слика 3.1.3.

```
ACTGCACCACA___CCC
|||  |||||||
ACTCCACCACAACT_CC
```

Слика 3.1.3. MUMmer порамнување

3.2. SSAHA индексирање и пребарување на ДНК база на податоци

SSAHA е пристап за пребарување на целосни и нецелосни совпаѓања по прашалник секвенца q во рамки на ДНК база на податоци $S=\{S_1,S_2,...,S_{n-1},S_n\}$. За да може да се пребарува базата на податоци, истата претходно се индексира. На почеток се конструира хеш-табела со 4^k покажувачи. Покажувач на положба $f(w_i)$, $0 \le f(w_i) \le 4^k-1$ соодветствува на еден од 4^k можни зборови $w_i = a_{i,1}a_{i,2}..a_{i,k-1}a_{i,k}$, над азбуката $\Sigma=\{A,C,T,G\}$, според шемата за конверзија (3.2.1), кодирајќи ја секоја база со вредност: 0, 1, 2 или 3.

f(A)=00_b=0

f(C)=01_b=1

f(G)=10_b=2

f(T)=11_b=3

$$f(w_i : a_{i,1}a_{i,2}..a_{i,k-1}a_{i,k}) = \sum_{j=1}^{k} f(a_{i,j}) \times 4^{j-1},\ a_{i,j} \in \{A,C,T,G\} \quad (3.2.1)$$

Покажувач на положба $f(w_i)$ покажува кон множество на податочни парови (x, y), каде $x, 1 \le x \le n$ е индекс на секвенца на припадност на непреклопувачки збор w_i, додека y е почетната положба на секое појавување на зборот w_i во рамки на секвенцата S_i.

За пример базата на податоци $S = \{S_1 : ACTGCC, S_2 = TCACCC, S_3 = AATCCG\}$, за $k = 2$, хеш табелата е прикажана на Слика 3.2.1. Од секоја секвенца се издвојуваат непреклопувачки зборови со должина $k = 2$: AC (1,1), TG (1,3), CC (1,5), TC (2,1), AC (2,3), CC (2,5), AA (3,1), TC (3,3) и CG (3,5). Паровите (x,y) ги означуваат секвенците на припадност на непреклопувачките зборови w_i и почетните положби на појавување на зборовите во рамки на секвенците. Така на пример зборот AC се појавува во рамки на секвенците S_1 и S_2. Почетната положба на зборот AC во S_1 е 1, додека почетната положба на истиот збор во S_2 е 3, релативно во однос на почетоците на секвенците.

Збор w	$f(w)$	(x, y)
AA	0	(3,1)
AC	1	(1,1) (2,3)
AG	2	
AT	3	
CA	4	
CC	5	(1,5), (2,5)
CG	6	(3,5)
CT	7	
GA	8	
GC	9	
GG	10	
GT	11	
TA	12	
TC	13	(2,1), (3,3)
TG	14	(1,3)
TT	15	

Слика 3.2.1. SSAHA хеш табела за пример секвенците

За да се најде совпаѓање на прашалник секвенца во рамки на базата на податоци, од прашалник секвенцата q се издвојуваат: $|q|-k+1$ преклопувачки зборови $q_i, 1 \le i \le |q|-k+1$. Ако за хеш вредноста на збор $q_i, f(q_i)$ постои барем една податочна двојка (x,y) во рамки на хеш табелата, тогаш се пресметува податочна тројка: $(x, y-t, y)$, каде t е почетната положба на зборот q_i, релативно во однос на почетокот на прашалник секвенцата. Применувајќи го

претходното за сите преклопувачки зборови q_i од прашалник секвенцата q, се образува листа на совпаѓања H.

На Слика 3.2.2 е дадена листата на совпаѓања H, ако базата на податоци се пребарува по прашалник секвенца q:TGCC. Вкупно $|q|-k+1=4-2+1=3$ преклопувачки зборови со должина 2 може да се издвојат од прашалник секвенцата: TG (0), GC(1) и CC(2). Броевите во загради ги означуваат почетните положби на издвоените зборови. Четиринаесеттиот покажувач од хеш табелата соодветствува на зборот TG. Истиот покажува кон податочна двојка (1,3), врз основа на што се пресметува податочна тројка во рамки на листата на погодоци H, (1, 3-0, 3)=(1, 3, 3). Покажувачот кој соодветствува на зборот GC не покажува кон податочна довојка, додека покажувачот кој соодветствува на зборот CC покажува кон две податочни двојки: (1,5) и (2,5), врз основа на што се пресметуваат уште две податочни тројки во рамки на листата на совпаѓања H: (1,5-2,5)=(1,3,5), (2,5-2,5)=(2,3,5).

Збор w	$f(w)$	(x,y)	Листа на совпаѓања H
AA	0	(3,1)	
AC	1	(1,1) (2,3)	
AG	2		
AT	3		
CA	4		
CC	5	(1,5), (2,5)	(1,3,5), (2,3,5)
CG	6	(3,5)	
CT	7		
GA	8		
GC	9		
GG	10		
GT	11		
TA	12		
TC	13	(2,1), (3,3)	
TG	14	(1,3)	(1,3,3)
TT	15		

Слика 3.2.2. Листа на совпаѓања H

Листата на совпаѓања H се сортира последователно по x, y-t и y вредностите, со што се добива сортирана листа M: (1,3,3), (1,3,5), (2,3,5). Целосните совпаѓања на прашалник секевнцата се определени со последователно множество на податочни тројки: (x_1, y_1-t_1, y_1)…($x_{|q|/k}$, $y_{|q|/k}$–$t_{|q|/k}$, $y_{|q|/k}$), така што важи: x_1=…=$x_{|q|/k}$, y_1-t_1=…=$y_{|q|/k}$-$t_{|q|/k}$.

За конкретниот пример, множеството на последователни тројки: (1,3,3), (1,3,5), определува целосно совпаѓање на прашалник секвенцата q:TGCC во рамки на секвенцата S_1, започнувајќи од положба 3.

4. Матрици на замени

4.1. BLOSUM матрица на замени

За конструкција на BLOSUM (Block Substitution Matrix) матрица на замени се порамнуваат повеќе сродни ДНК или протеински секвенци, употребувајќи метрика на порамнување определена преку единечна матрица на сличност. Матрицата на Слика 4.1.1, може да се земе како пример единечна матрица на сличност, каде резултатот за порамнување на идентични нуклеотиди изнесува 1, додека резултатот за порамнување на нуклеотид со празнина или различни нуклеотиди изнесува 0.

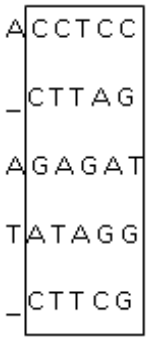
	A	C	T	G	_
A	1	0	0	0	0
C	0	1	0	0	0
T	0	0	1	0	0
G	0	0	0	1	0
-	0	0	0	0	

Слика 4.1.1. Единечна матрица на сличност

Од повеќекратното порамнување се издвојува фрагмент на безпразнинско порамнување со висок резултат на порамнување. Резултатот на порамнување се пресметува како збир на резултати на различни комбинации на единечни парови порамнувања, за секоја колона од фрагментот на безпразнинско повеќекратно порамнување. Резултатот на порамнување за безпразнинскиот фрагмент на Слика 4.1.2 се пресметува како: $s = s_{col1} + s_{col2} + s_{col3} + s_{col4} + s_{col5}$.

Резултатот на порамнување по првата колона изнесува: $s_{col1} = s(C,C) + s(C,G) + s(C,A) + S(C,C) + s(C,G) + s(C,A) + s(C,C) + s(G,A) + s(G,C) + s(A,C) = 3$. На ист начин се пресметуваат резултатите на порамнување по останатите колони: $s_{col2} = 3$,

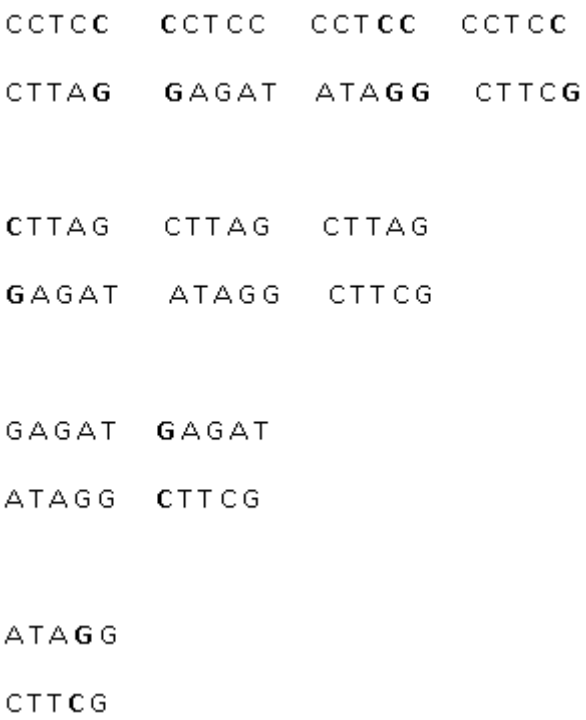
$s_{col3}=3$, $s_{col4}=2$, $s_{col5}=3$. Резултатот на порамнување за безпразнинскиот фрагмент изнесува: $s=3+3+3+2+3=14$.



Слика 4.1.2. Фрагмент на безпразнинско порамнување

Од издвоениот фрагмент на безпразнинско порамнување може да се пресметат фрекфенциите на појавување на секој карактер: $f_x=\frac{n_x}{n}$, каде n_x е број на појавувања на карактерот x во рамки на фрагментот, додека n е вкупен број на карактери во рамки на фрагментот на безпразнинско повеќекратно порамнување. За конкретниот пример: $f_A=\frac{5}{25}$, $f_G=\frac{6}{25}$, $f_C=\frac{7}{25}$, $f_T=\frac{7}{25}$. Нуклеотидот A се појавува пет пати, нуклеотидот G се појавува шест пати, додека нуклеотидите C и T се појавуваат по седум пати.

Издвојувајќи ги комбинациите на парови порамнувања од безпразнинскиот фрагмент на порамнување, може да се преметат набљудуваните фрекфенции: $f_{o,x\leftrightarrow y}=\frac{n_{o,x\leftrightarrow y}}{n_o}$, каде $n_{o,x\leftrightarrow y}$ е вкупен број на $x\leftrightarrow y$ парови субституции, n_o е вкупен број на парови порамнувања. За примерот што се разгледува, бројот на $C\leftrightarrow G$ субституции изнесува 8, Слика 4.1.3. Бидејќи вкупниот број на единечни парови порамнувања изнесува 50, $f_{o,C\leftrightarrow G}$ се пресметува како: $f_{o,C\leftrightarrow G}=\frac{8}{50}$.



Слика 4.1.3. Приказ на C↔G базни субституции

$f_{o,C\leftrightarrow C} = 4/50,\ f_{o,C\leftrightarrow T} = 4/50,\ f_{o,C\leftrightarrow A} = 8/50,\ f_{o,C\leftrightarrow G} = 8/50$

$f_{o,T\leftrightarrow A} = 6/50,\ f_{o,T\leftrightarrow T} = 6/50,\ f_{o,T\leftrightarrow G} = 6/50$

$f_{o,G\leftrightarrow A} = 4/50,\ f_{o,G\leftrightarrow G} = 3/50$

$f_{o,A\leftrightarrow A} = 1/50$

Откако ќе се пресметат сите набљудувани фрекфенции, вредностите: $2\log_2 \frac{f_{o,x\leftrightarrow y}}{f_x \times f_y}$ се додаваат во рамки на симетрична BLOSUM $b[x,y]$матрица, каде важи: $b[x,y] = b[y,x] = 2\log_2 \frac{f_{o,x\leftrightarrow y}}{f_x \times f_y}$. Бројот на редици и колони во рамки на BLOSUM матрицата е еднаков и истиот соодветствува на бројот на различни карактери, кои се појавуваат во рамки на безпразнинскиот фрагмент на повеќекратно порамнување. На Слика 4.1.4 е прикажана BLOSUM матрицата за разгледуваниот пример. Полето $b[C,C]$ се пресметува како: $b[C,C] = 2\log_2\left(\frac{f_{o,C\leftrightarrow C}}{f_C \times f_C}\right) = 2\log_2\left(\frac{4/50}{7/25 \times 7/25}\right) = 0{,}0582$. Претходната вредност се заокружува до најблиската целобројна вредност 0. На ист начин се пресметуваат и останатите полиња од BLOSUM матрицата.

	C	T	G	A
C	**0**	**0**	**3**	**3**
T	**0**	**1**	**2**	**2**
G	**3**	**2**	**0**	**1**
A	**3**	**2**	**1**	**-2**

Слика 4.1.4. BLOSUM матрица

$$b[C,T] = 2\log_2\left(\frac{f_{o,C\leftrightarrow T}}{f_C \times f_T}\right) = 2\log_2\left(\frac{4/50}{7/25\times 7/25}\right) = 0{,}0582$$

$$b[C,G] = 2\log_2\left(\frac{f_{o,C\leftrightarrow G}}{f_C \times f_G}\right) = 2\log_2\left(\frac{8/50}{7/25\times 6/25}\right) = 2{,}503$$

$$b[C,A] = 2\log_2\left(\frac{f_{o,C\leftrightarrow A}}{f_C \times f_A}\right) = 2\log_2\left(\frac{8/50}{7/25\times 5/25}\right) = 3{,}029$$

$$b[T,T] = 2\log_2\left(\frac{f_{o,T\leftrightarrow T}}{f_T \times f_T}\right) = 2\log_2\left(\frac{6/50}{7/25\times 7/25}\right) = 1{,}228$$

$$b[T,G] = 2\log_2\left(\frac{f_{o,T\leftrightarrow G}}{f_T \times f_G}\right) = 2\log_2\left(\frac{6/50}{7/25\times 6/25}\right) = 1{,}673$$

$$b[T,A] = 2\log_2\left(\frac{f_{o,T\leftrightarrow A}}{f_T \times f_A}\right) = 2\log_2\left(\frac{6/50}{7/25\times 5/25}\right) = 2{,}199$$

$$b[G,G] = 2\log_2\left(\frac{f_{o,G\leftrightarrow G}}{f_G \times f_G}\right) = 2\log_2\left(\frac{3/50}{6/25\times 6/25}\right) = 0{,}119$$

$$b[G,A] = 2\log_2\left(\frac{f_{o,G\leftrightarrow A}}{f_G \times f_A}\right) = 2\log_2\left(\frac{4/50}{6/25\times 5/25}\right) = 1.475$$

$$b[A,A] = 2\log_2\left(\frac{f_{o,A\leftrightarrow A}}{f_A \times f_A}\right) = 2\log_2\left(\frac{1/50}{5/25\times 5/25}\right) = -2$$

4.2. PAM матрица на замени

PAM (Point Accepted Mutation) е модел на замена на една амино-киселина со друга, *прифатлива од аспект на природна селекција*. Мутацијата е прифатлива од аспект на природна селекција ако истата не е причина за изумирање на конкретен организам. Димензиите на PAM матрицата се 20x20, каде секоја редица (колона) соодветствува на една протеинска амино-киселина. Вредностите од PAM матрицата на замени се земат како метрика на порамнување при порамнување на протеински секвенци.

За да се конструира PAM матрица се порамнуваат повеќе протеински секвенци, со процент на идентичност од најмалку 85%, Слика 4.2.1. За секоја амино-киселина i се пресметува релативната мутабилност m_i, како колични помеѓу бројот на $i \to j$ субституции и бројот на појавувања на амино-киселината i. На Слика 4.2.1, амино-киселината аланин (A) се заменува со аланин или друга амино-киселина 28 пати. Аланинот се појавува 10 пати во рамки на повеќекратното порамнување, од каде за релативната мутабилност за амино-киселината аланин (A) се добива $m_A = \frac{28}{10} = 2{,}8$.

Во наредниот чекор се конструира филогенетско стебло за секвенците од повеќекратното порамнување, Слика 4.2.2, од каде може да се пресметат веројатностите за замена на амино-киселина j со амнино-киселина i $M_{i,j} = \frac{m_j F_{i,j}}{\sum_i F_{i,j}}$, каде m_j е релативната мутабилост за амино-киселината j, $F_{i,j}$ е број на субституции на амино-киселината j со амино-киселина i, додека $\sum_i F_{i,j}$ е број на субституции на амино-киселината j со друга амино-киселина. За примерот што се разгледува, бројот на замени $A \to G$ ($G \to A$) изнесува 3, аланинот се заменува со друга амино-киселина 4 пати, додека мутабилноста за аланин изнесува 2,8. Врз основа на претходните вредности за $M_{G,A}$ се добива: $M_{G,A} = \frac{2{,}8 \times 3}{4} = 2{,}1$.

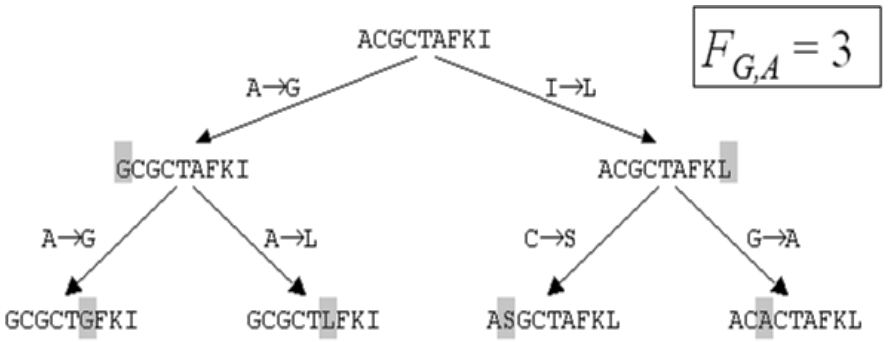
Се пресметуваат и вредностите f_i, како количник помеѓу бројот на појавувања на амино-киселината i и бројот на амино-киселини во рамки на порамнувањето. За конкретниот пример, амино-киселината G се поајува 10 пати, додека вкупниот број на амино-киселини изнесува 63, од каде за f_G се добива $f_G = \frac{10}{63} = 0{,}1587$. Вредноста на полето $PAM_{i,j}$ ($PAM_{j,i}$) (за амино-киселина i по редица и амино-киселина j по колона и обратно) се пресметува како $PAM_{i,j} = \log\left(\frac{M_{i,j}}{f_i}\right)$.

За $PAM_{G,A}$, земајќи ги претходните вредности се добива :

$$PAM_{G,A} = \log\left(\frac{M_{G,A}}{f_G}\right) = \log\left(\frac{2{,}1}{0{,}1587}\right) = \log(12{,}76) = 1{,}106\,.$$

На ист начин се пресметуваат и останатите полиња од PAM матрицата.

Слика 4.2.1. Фрагмент на порамнување, каде процентот на идентичност изнесува најмалку 85%



Слика 4.2.2. Приказ на филогенетско стебло

4.3. Еволуциски модел на Jukes и Cantor

ДНК еволуцијата е стохастички процес. Марковите процеси се погодни за моделирање на процесот на ДНК еволуција. Меѓусебно овие модели се разликуваат по параметрите за опис на ратата на нуклеотидни замени.

Наједноставен ДНК еволуциски модел е симетричниот модел на Jukes и Cantor, каде е подеднаква веројатноста за супституција на база x со друга база y, $x \neq y$ во рамки на единечен временски интервал (од t во $t+1$) . Во рамки на меделот на Jukes и Cantor фигурира само еден параметар α, што претставува *рата на набљудувани базни супституции*. Така на пример за митохондријална ДНК, α изнесува 10^{-8} (10^{-8} базни замени по позиција за година).

На Слика 4.3.1 е дадена Jukes-Cantor матрицата на веројатности на базни замени. Веројатноста за замена на нуклеотид x со нуклеотид y, $P_{x,y}$, $x \neq y$ во рамки на единечен временски интервал изнесува $\frac{\alpha}{3}$, додека веројатноста база x да не се замени со друга база $P_{x,x}$ изнесува $1-\alpha$.

$$M = \begin{bmatrix} P_{A,A}:1-\alpha & P_{A,C}:\frac{\alpha}{3} & P_{A,G}:\frac{\alpha}{3} & P_{A,T}:\frac{\alpha}{3} \\ P_{C,A}:\frac{\alpha}{3} & P_{C,C}:1-\alpha & P_{C,G}:\frac{\alpha}{3} & P_{C,T}:\frac{\alpha}{3} \\ P_{G,A}:\frac{\alpha}{3} & P_{G,C}:\frac{\alpha}{3} & P_{G,G}:1-\alpha & P_{G,T}:\frac{\alpha}{3} \\ P_{T,A}:\frac{\alpha}{3} & P_{T,C}:\frac{\alpha}{3} & P_{T,G}:\frac{\alpha}{3} & P_{T,T}:1-\alpha \end{bmatrix}$$

Слика 4.3.1. Jukes-Cantor матрица на веројатности на базни замени

Матрицата M може да се запише како: $M = \left(1-\frac{4}{3}\alpha\right)I + \frac{4}{3}\alpha J$, каде:

$$I = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad J = \begin{bmatrix} \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \end{bmatrix}$$

Со замената $r = 1-\frac{4}{3}\alpha$, равенството за M може да се запише како: $M = rI + (1-r)J = K(r)$. За било кои два реални броеви: r и s, може да се пресмета производот $K(r)K(s)$ - равенство (4.3.1). Забележете дека важи: $I \cdot I = I$ и $I \cdot J = J \cdot I = J \cdot J = J$.

$$K(r)K(s) = (rI + (1-r)J)(sI + (1-s)J) = rsI + (r(1-s) + s(1-r) + (1-r)(1-s))J = rsI + (1-rs)J = K(rs) \quad (4.3.1)$$

Од равенството (4.3.1) се добива дека важи: $M^t = K(r)^t = K(r^t) = r^t I + (1-r^t)J$, каде M^t е матрица, која ги определува веројатностите за замена на нуклеотид x со нуклеотид y после t единечни временски интервали.

$$M^t = \left(1-\frac{4}{3}\alpha\right)^t \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} + \left(1-\left(1-\frac{4}{3}\alpha\right)^t\right) \begin{bmatrix} \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \end{bmatrix} =$$

$$= \begin{bmatrix} \frac{1}{4}+\frac{3}{4}\left(1-\frac{4}{3}\alpha\right)^t & \frac{1}{4}-\frac{1}{4}\left(1-\frac{4}{3}\alpha\right)^t & \frac{1}{4}-\frac{1}{4}\left(1-\frac{4}{3}\alpha\right)^t & \frac{1}{4}-\frac{1}{4}\left(1-\frac{4}{3}\alpha\right)^t \\ \frac{1}{4}-\frac{1}{4}\left(1-\frac{4}{3}\alpha\right)^t & \frac{1}{4}+\frac{3}{4}\left(1-\frac{4}{3}\alpha\right)^t & \frac{1}{4}-\frac{1}{4}\left(1-\frac{4}{3}\alpha\right)^t & \frac{1}{4}-\frac{1}{4}\left(1-\frac{4}{3}\alpha\right)^t \\ \frac{1}{4}-\frac{1}{4}\left(1-\frac{4}{3}\alpha\right)^t & \frac{1}{4}-\frac{1}{4}\left(1-\frac{4}{3}\alpha\right)^t & \frac{1}{4}+\frac{3}{4}\left(1-\frac{4}{3}\alpha\right)^t & \frac{1}{4}-\frac{1}{4}\left(1-\frac{4}{3}\alpha\right)^t \\ \frac{1}{4}-\frac{1}{4}\left(1-\frac{4}{3}\alpha\right)^t & \frac{1}{4}-\frac{1}{4}\left(1-\frac{4}{3}\alpha\right)^t & \frac{1}{4}-\frac{1}{4}\left(1-\frac{4}{3}\alpha\right)^t & \frac{1}{4}+\frac{3}{4}\left(1-\frac{4}{3}\alpha\right)^t \end{bmatrix} =$$

$$= \begin{bmatrix} P^{(t)}{}_{A,A} & P^{(t)}{}_{A,C} & P^{(t)}{}_{A,G} & P^{(t)}{}_{A,T} \\ P^{(t)}{}_{C,A} & P^{(t)}{}_{C,C} & P^{(t)}{}_{C,G} & P^{(t)}{}_{C,T} \\ P^{(t)}{}_{G,A} & P^{(t)}{}_{G,C} & P^{(t)}{}_{G,G} & P^{(t)}{}_{G,T} \\ P^{(t)}{}_{T,A} & P^{(t)}{}_{T,C:} & P^{(t)}{}_{T,G} & P^{(t)}{}_{T,T} \end{bmatrix}$$

Веројатноста за замена на база x со база y после t временски интервали изнесува: $P^{(t)}x, y = \frac{1}{4}-\frac{1}{4}\left(1-\frac{4}{3}\alpha\right)^t$, додека веројатноста да не се случи базна замена после t временски интервали изнесува: $P^{(t)}x, x = \frac{1}{4}+\frac{3}{4}\left(1-\frac{4}{3}\alpha\right)^t$.

На пример, веројатноста за замена на нуклеотидот аденин со нуклеотидот тимин, после 50 временски интервали изнесува: $P^{(50)}A, T = \frac{1}{4}-\frac{1}{4}\left(1-\frac{4}{3}\alpha\right)^{50}$, додека веројатноста да не се случи базна замена после 50 временски интервали се пресметува како: $P^{(50)}A, A = \frac{1}{4}+\frac{3}{4}\left(1-\frac{4}{3}\alpha\right)^{50}$. Ако веројатностите за базна замена се пресметуваат за

митохондријална ДНК, каде $\alpha = 10^{-8}$, тогаш за претходните веројатности се добива: $P^{(50)}A,T = \frac{1}{4} - \frac{1}{4}\left(1 - \frac{4\times10^{-8}}{3}\right)^{50}$ и $P^{(50)}A,A = \frac{1}{4} + \frac{3}{4}\left(1 - \frac{4\times10^{-8}}{3}\right)^{50}$.

4.4. Еволуциски модел на Kimura со два параметри

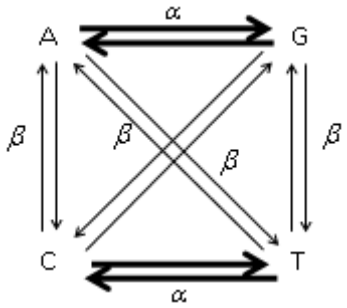
За разлика од моделот на Jukes и Cantor, моделот на Kimura е еволуциски модел, кој вклучува 2 параметри α и β, каде α претставува *рата на базни транзиции*, додека β претставува *рата на базни трансверзии* по единечна нуклеотидна положба во рамки на единечен временски интервал. Транзиција претставува базна замена помеѓу два пиримидини $(T \leftrightarrow C)$ или два пурини $(A \leftrightarrow G)$, додека трансверзија е базна замена помеѓу пиримидин и пурнин, и обратно $(TC \leftrightarrow AG)$. Во реалноста транзициите се случуваат почесто од трансверзиите, па затоа неопходно е да се земат различни вредности за веројатност за транзиција и трансверзија на база во рамки на единечен временски интервал (различни вредности за α и β).

Матрицата на веројатности на базни замени во рамки на единечен временски интервал е дадена на Слика 4.4.1. Веројатноста за базна транзиција изнесува α, веројатноста за базна транзверзија изнесува β, додека веројатноста да не се случи базна замена во рамки на единечен временски интервал изнесува $1-\alpha-2\beta$.

Веројатноста да не се случи базна замена после t временски интервали $P_{x,x}^{(t)}$ се пресметува како: $P_{x,x}^{(t)} = \frac{1}{4} + \frac{1}{4}e^{-4\beta t} + \frac{1}{2}e^{-2(\alpha+\beta)t}$, веројатноста за базна транзиција после t единечни временски интервали $P_{transition}^{(t)}$ се пресметува како: $P_{transition}^{(t)} = \frac{1}{4} + \frac{1}{4}e^{-4\beta t} - \frac{1}{2}e^{-2(\alpha+\beta)t}$ и веројатноста за базна трансверзија после t единечни временски интервали се пресметува како: $P_{transversion}^{(t)} = \frac{1}{4} - \frac{1}{4}e^{-4\beta t}$.

$$M = \begin{bmatrix} P_{A,A}:1-\alpha-2\beta & P_{A,C}:\beta & P_{A,G}:\alpha & P_{A,T}:\beta \\ P_{C,A}:\beta & P_{C,C}:1-\alpha-2\beta & P_{C,G}:\beta & P_{C,T}:\alpha \\ P_{G,A}:\alpha & P_{G,C}:\beta & P_{G,G}:1-\alpha-2\beta & P_{G,T}:\beta \\ P_{T,A}:\beta & P_{T,C:}:\alpha & P_{T,G}:\beta & P_{T,T}:1-\alpha-2\beta \end{bmatrix}$$

Слика 4.4.1. Матрица на веројатности на базни замени



Слика 4.4.2. Шематски приказ на Kimura еволуцискиот модел

5. Филогенетска анализа, UPGMA и метод на Fitch и Margoliash

5.1. UPGMA

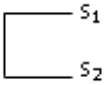
UPGMA (Unweighted Pair Group Method) е метод за конструкција на филогенетско стебло. Филогенетското стебло ја прикажува еволуциската релација помеѓу биолошките организми. Примената на UPGMA методот ќе ја покажеме за пример секвенците: s_1: AGCCT, s_2: ACCCT, s_3: ATGGT, s_4: ACCTT и s_5: AAAAG.

За секој пар секвенци s_i,s_j се пресметува растојание $d_{i,j}$ како количник помеѓу број на несовпаѓања и должина на порамнување. За секвенците s_1 и s_2, растојанието $d_{1,2}$ изнесува $d_{i,j}$=1/5=0,2 (едно базно несовпаѓање од пет базни порамнувања). На ист начин се пресметуваат и останатите растојанија.

	1	2	3	4	5
1					
2	0,2				
3	0,6	0,6			
4	0,4	0,2	0,6		
5	0,8	0,8	0,8	0,8	

Слика 5.1.1. UPGMA матрица на растојанија

Секвенците s_i и s_j, за кои важи дека растојанието $d_{i,j}$ е минимално формираат заеднички кластер. За пример секвенците, растојанието $d_{1,2}$ е минимално, Слика 5.1.2.



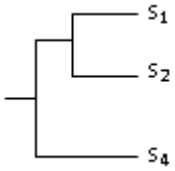
Слика 5.1.2. Секвенците s_1 и s_2 формираат заеднички кластер

Во наредниот чекор се конструира изменета матрица на растојанија. Истата вклучува и растојанија помеѓу некластерираните секвенци и веќе кластерираните секвенци. Така на пример, растојанието помеѓу некластерираната секвенца s_3 и С: s_1,s_2 кластерот се пресметува како: $d_{3,C}=(d_{3,1}+d_{3,2})/2=(0,6+0,6)/2=0,6$. На ист начин се пресметуваат и растојанијата: $d_{4,C}=(d_{4,1}+d_{4,2})/2=(0,4+0,2)/2=0,3$ и $d_{5,C}=(d_{5,1}+d_{5,2})/2=(0,8+0,8)/2=0,8$, Слика 5.1.3.

	1,2	3	4	5
1,2				
3	0,6			
4	0,3	0,6		
5	0,8	0,8	0,8	

Слика 5.1.3. Изменета матрица на растојанија

Од сите растојанија, најмало е растојанието помеѓу секвенцата s_4 и кластерот С: s_1,s_2. Како резултат на тоа, кон кластерот С се додава секвенцата s_4, С: s_1,s_2,s_4, Слика 5.1.4.



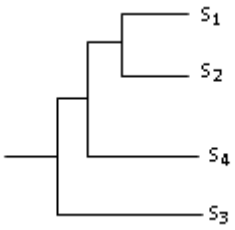
Слика 5.1.4. Додавање на секвенцата s_4

По аналогија на претходното, се пресметуваат растојанијата помеѓу секвенците: s_3, s_5 и кластерот С, со што се образува изменета матрица на растојанија, Слика 5.1.5. Растојанието помеѓу секвенцата s_3 и кластерот С: s_1,s_2,s_4 се пресметува како: $d_{3,C}=(d_{3,1}+d_{3,2}+d_{3,4})/3=(0,6+0,6+0,6)/3=0,6$. На ист начин се пресметува и растојанието помеѓу секвенцата s_5 и кластерот С: s_1,s_2,s_4 : $d_{5,C}=(d_{5,1}+d_{5,2}+d_{5,4})/3=(0,8+0,8+0,8)/3=0,8$.

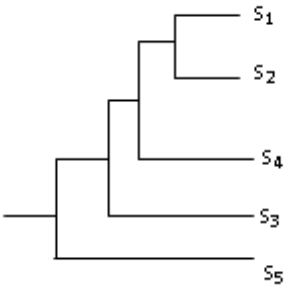
Од изменетата матрица на растојанија, најмало е растојанието помеѓу секвенцата s_3 и кластерот С: s_1,s_2,s_4, што резултира со групирање на s_3 кон С, С: s_1,s_2,s_4,s_3, Слика 5.1.6. На крај се групира секвенцата s_5, Слика 5.1.7.

	1,2,4	3	5
1,2,4			
3	0,6		
5	0,8	0,8	

Слика 5.1.5. Изменета матрица на растојанија



Слика 5.1.6. Додавање на секвенцата s_3

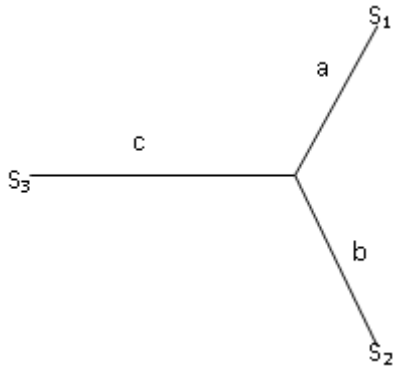


Слика 5.1.7. UPGMA филогенетско стебло

5.2. Метод на Fitch и Margoliash

Со примена на методот на Fitch и Margoliash се конструира безкоренесто филогенетско стебло. Како растојание помеѓу две секвенци s_i и s_j се зема бројот на базни разлики. Збирот на растојанија помеѓу јазлите j_i и j_j е растојание помеѓу секвенците s_i и s_j. Должината на секоја гранка од филогенетското стебло претставува растојание.

На Слика 5.2.1, е дадено пример филогенетско стебло според Fitch и Margoliash за секвенците s_1, s_2 и s_3. Растојанието помеѓу секвенците s_1 и s_2 изнесува $d_{1,2}$ ($a+b=d_{1,2}$), растојанието помеѓу секвенците s_1 и s_3 изнесува $d_{1,3}$ ($d_{1,3}=a+c$) и растојанието помеѓу секевнците s_2 и s_3 изнесува $d_{2,3}$ ($d_{2,3}=b+c$). Со решевање на систем равенки од три равенки, со три непознати (5.2.1), може да се определат должините на гранките: a, b и c (5.2.2,5.2.3,5.2.4).



Слика 5.2.1. Безкоренесто филогенетско стебло

$$\begin{cases} a+b=d_{1,2} \\ a+c=d_{1,3} \\ b+c=d_{2,3} \end{cases} \qquad (5.2.1)$$

$$a=\left(d_{1,3}+d_{1,2}-d_{2,3}\right)/2 \qquad (5.2.2)$$

$$b=\left(d_{1,2}+d_{2,3}-d_{1,3}\right)/2 \qquad (5.2.3)$$

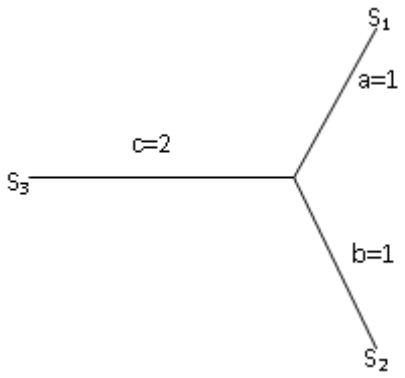
$$c=\left(d_{1,3}+d_{2,3}-d_{1,2}\right)/2 \qquad (5.2.4)$$

Така на пример, ако растојанието помеѓу секвенците s_1 и s_2 изнесува $d_{1,2}$=2, растојанието помеѓу секвенците s_1 и s_3 изнесува $d_{1,3}$=3 и растојанието помеѓу секвенците s_2 и s_3 изнесува $d_{2,3}$=3, за должините на гранките според (5.2.2, 5.2.3 и 5.2.4) се добива: $a=1, b=1, c=2$, Слика 5.2.2.

$$a=\left(d_{1,3}+d_{1,2}-d_{2,3}\right)/2=\left(3+2-3\right)/2=1$$

$$b=\left(d_{1,2}+d_{2,3}-d_{1,3}\right)/2=\left(2+3-3\right)/2=1$$

$$c=\left(d_{1,3}+d_{2,3}-d_{1,2}\right)/2=\left(3+3-2\right)/2=2$$



Слика 5.2.2. Безкоренесто филогенетско стебло за растојанијата: $d_{1,2}$, $d_{1,3}$ и $d_{2,3}$

Примената на методот на Fitch и Margoliash ќе ја покажеме за пример секвенците: s_1:AAAAA, s_2: AACCA, s_3: AATTT и s_4: ATGGG. Постапката започнува со конструкција на симетрична матрица на растојанија, каде $d_{i,j}$($d_{i,j}=d_{j,i}$) е растојание помеѓу секвенците s_i и s_j (број на базни разлики). Вредноста на полето $s_{1,2}$ изнесува 2, бидејќи бројот на базни разлики помеѓу секвенците s_1 и s_2 изнесува 2. По аналогија на претходното се пресметуваат и останатите полиња од симетричната матрица на растојанија, Слика 5.2.3.

s_1:AAAAA

s_2: AACCA, $d_{1,2}$=2

s_1:AAAAA

s_3: AATTT, $d_{1,3}$=3

s_1:AAAAA

s_4: ATGGG, $d_{1,4}$=4

s_2: AACCA

s_3: AATTT, $d_{2,3}$=3

s_2: AACCA

s_4: ATGGG, $d_{2,4}$=4

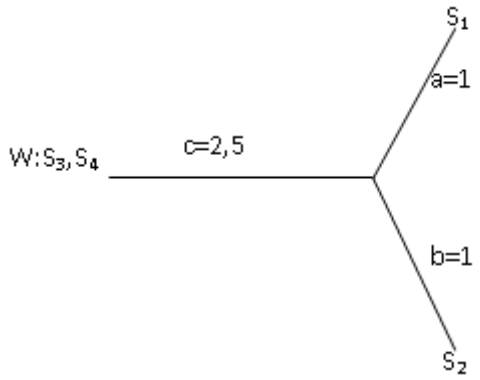
s_3: AATTT

s_4: ATGGG, $d_{3,4}$=4

	S_1	S_2	S_3	S_4
S_1				
S_2	2			
S_3	3	3		
S_4	4	4	4	

Слика 5.2.3. Матрица на растојанија

Со избор на најмалку одалечените секвенци, s_1 и s_2 и групирање на остатокот секвенци, s_3 и s_4, се конструира филогенетско стебло како на Слика 5.2.4. За да се најдат должините на гранките a,b и c, неопходно е да се најдат растојанијата: $d_{1,2}$, $d_{1,W}$ и $d_{2,W}$, каде $d_{1,W}$ и $d_{2,W}$ се растојанија помеѓу секвенците s_1 и s_2 и кластерот C: s_3,s_4, кои се пресметуваат како: $d_{1,W}$ =($d_{1,3}$+$d_{1,4}$)/2=(3+4)/2=3.5 и $d_{2,W}$ =($d_{2,3}$+$d_{2,4}$)/2=(3+4)/2=3.5.



Слика 5.2.4. Приказ на филогенетско стебло

За да се најдат должините на гранките: a,b и c, треба да се реши системот од три равенки, со три непознати (5.2.5), од каде се добива: a=1, b=1 и c=2,5, Слика 5.2.4.

$$\begin{cases} a+b=d_{1,2}=2 \\ a+c=d_{1,W}=3{,}5 \\ b+c=d_{2,W}=3{,}5 \end{cases} \quad (5.2.5)$$

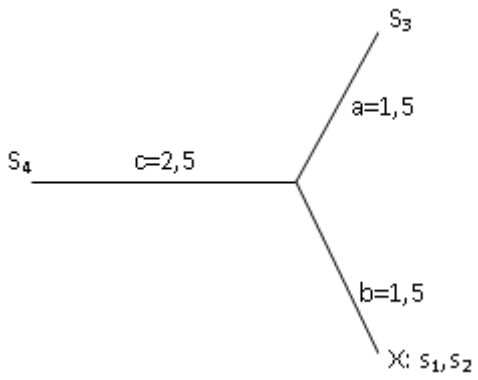
Во наредниот чекор се конструира изменета матрица на растојанија, која ги вклучува растојанијата помеѓу секвенците од кластерот W: s_3,s_4 и кластерот на секвенци на најмала оддалечност, X: s_1 и s_2, Слика 5.2.5. Растојанијата $d_{3,X}$ и $d_{4,X}$ се пресметуваат на начин: $d_{3,X}=(d_{3,1}+d_{3,2})/2=(3+3)/2=3$ и $d_{4,X}=(d_{4,1}+d_{4,2})/2=(4+4)/2=4$.

	S_3	S_4	X: S_1,S_2
S_3			
S_4	4		
X: S_1,S_2	3	4	

Слика 5.2.5. Матрица на растојанија

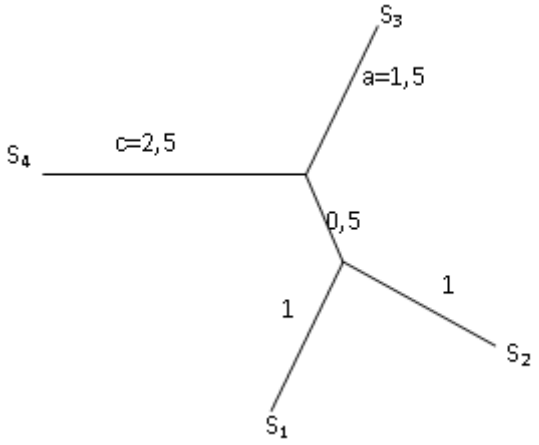
Со избор на полето со минимална вредност од изменетата матрица на растојанија, полето со вредност 3, се конструира филогенетското стебло на Слика 5.2.6. За да се пресметат должините на гранките: a, b и c, се решава системот равенки (5.2.6), од каде се добива: a=1,5, b=1,5 и c=2,5.

$$\begin{cases} a+b=d_{3,X}=3 \\ a+c=d_{3,4}=4 \\ b+c=d_{4,X}=4 \end{cases} \quad (5.2.6)$$



Слика 5.2.6. Приказ на филогенетско стебло

Додавајќи го разгранувањето помеѓу секевцните s_1 и s_2 од претходно, се образува конечното филогенетско стебло за пример секвенците: s_1,s_2,s_3 и s_4, Слика 5.2.7.



Слика 5.2.7. Филогенетско стебло според Fitch и Margoliash

6. Методи за моделирање и предвидување на просторната структурата на протеините

Постојат два пристапи за моделирање на просторната структура на протеините. Тоа се: *компаративното моделирање* и *de novo пристапите*. Компаративното моделирање ја предвидува просторната структура на протеин, споредувајќи ја неговата примарна структура со примарните структури на протеини со позната просторна структура. Се очекува, протеини со слична примарна структура да имат и слична просторна структура. Ако процентот на идентичност на примарно ниво помеѓу референтниот протеин, чија просторна структура е позната, со протеинот, чија просторна структура се предвидува, изнесува најмалку 50%, тогаш точноста на предвидениот просторен модел е задоволителна. Како алгоритам за наоѓање на протеини со висок процент на идентичност на примарно ниво во однос на протеинска секвенца, чија просторна структура се предвидува, најчесто се користи BLAST. Просторните

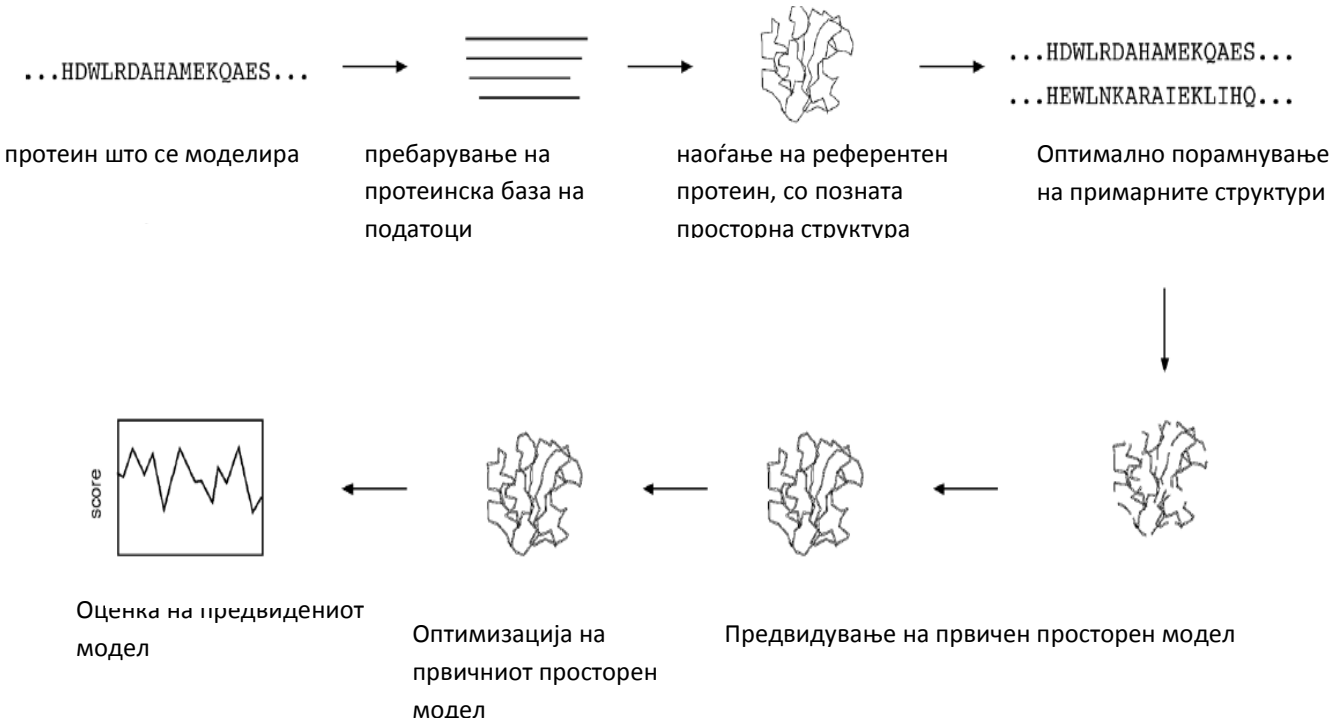
структури на споредбените протеини, по кои пребарува BLAST, се експериментално утврдени и точно познати. Кога резултатот од пребарувањето се повеќе протеински секвенци, тогаш просторната структура на протеинот со најголем процент на идентичност на примарно ниво, се зема како модел за градба на просторниот модел на протеинот со позната примарна структура, но непозната просторна структура.

За разлика од компаративното моделирање, de novo пристапите немат потреба од архива на референтни протеини со познати просторни структури, врз основа на што се врши предвидување на просторната структура на протеин, што ги прави применливи за секој протеин. Некои de novo пристапи ја предвидуваат просторната структура на протеин врз основа на енергентскиот баланс во внатрешноста на протеинот. Како типичен претставник, ќе ги издвоиме *ab initio* методите, кои вршат предвидување врз основа на енергетски функции, кои ги опишуваат хемиско-физичките интеракции помеѓу атомите во рамки на протеиниот. Поради комплексните математички пресметки, ab initio методите се покажаа како применливи за предвидување на просторната структура на кратки сегементи од протеински секвенци. Комплексноста на математичките пресметки, ја ограничува нивната примена на долги протеински секвенци. И покрај огромната точност на поновите ab initio методи при предвидување на просторната структура на пократки протеински секвенци и сегменти од подолги протеини, истите се неприменлви за подолги протеински секвенци, што укажува на предност на компаративните методи во однос на de novo методите.

6.1. Хомологно моделирање

Хомологното моделирање е процес што се одвива во повеќе фази. Првин се пребарува протеинска база на податоци за да се најдат хомологни протеини со познати просторни структури. Се избираат протеини со најмалку 30% идентичност на примарно ниво со протеинот, чија просторна структура се предвидува. Во случај на постоење на повеќе протеини, со процент на идентичност на примарно ниво од најмалку 30%, се избира протеинот со највисок процент на идентичност на примарно ниво со протеинот, чија просторна структура се предвидува, по што следи процес на оптимално порамнување на примарните структури на избраниот протеински модел од базата на податоци, чија просторна структура е позната, со примарната структура на протеинот, предмет на просторно моделирање. Протеините оптимално се порамнуваат за да се најде максимален можен број на еквивалентни резидиумски парови, по што координатите на евивалентно порамнетите остатоци од референтниот просторен протеински модел се копираат во просторот каде се врши

предвидување на структурата на предметниот протеин. Координатите на неквивалентно порамнетите остатоци, се утврдуваат со анализа на взаемните дејствијата помеѓу соседните остатоци. Имајќи во предвид дека просторната структура на протеинот е резултат на минимален енергетски баланс помеѓу атомите, во фазата на оптимизација на првично предвидениот протеински просторен модел, дел од првично предвидените протеински координати се корегираат, во правец на задоволување на условот за минимален меѓу-атомски енергетски баланс. Завршно, предвидениот просторен модел се оценува, преку проверка на параметри, како: растојанија помеѓу атомите, просторни агли,...итн.



Слика 6.1 Хомологно моделирање на протеини

6.2. AB INITO моделирање

За да се предвиди просторната структура на протеин со примена на хомологно моделирање, неопходно е да постои барем еден хомологен протеин со позната просторна структура за протеинот, чија просторна структура се предвидува. Ако пребарувањето за хомологни протеини не даде ниту еден позитивен резултат, тогаш хомологното моделирање е целосно неприменливо.

За разлика од хомологното моделирање, ab initio моделирањето нема потреба од архива на познати просторни модели на протеини. Предвидувањето се врши врз основа на познати просторни извиткувања на сегементи од протеини. Како типични преставници на ab inito пристапите ќе ги идвоиме алгоритмот на *Chou и Fasman* и методот на *Gor*.

Едни амино-киселини се начесто дел од алфа-хеликс, додека други дел од бета рамнина. Така на пример, аланинот, глутаминската киселина и метионинот се најчесто дел од алфа-хеликс, додека мала е веројатноста глицинот и пролинот да бидат составен дела на алфа-хеликс извиткување. Базиран на претходната дискусија, алгоритмот на Chou и Fashman врши предвидување на секундарните извиткувања, врз основа на очекувањето за припадност на конкретни амино-киселини кон познати секундарни извиткувања.

Методот на Gor врши предвидување врз основа на припадност на протеински подсекевнци со должина од 17 амино-киселини кон едно од четирите секундарни извиткувања: хеликс, стандард, извиткување и навивка.

6.3. Компјутерска визуелизација на протеини

Секоја компјутерска програма за визуелизација на структура на протеин, поддржува еден или повеќе различни молекуларни визуелизациски формати. Молекуларниот визуелизациски формат воспоставува релација помеѓу протеинските атоми и истиот овозможува приказ на протеинската молекула од одреден аспект на разгледување. Пософистицираните програми за визуелизација, интегрираат повеќе молекуларни визуелизациски формати, со што се озвозможува приказ на структурата на протеинот од различни аспекти.

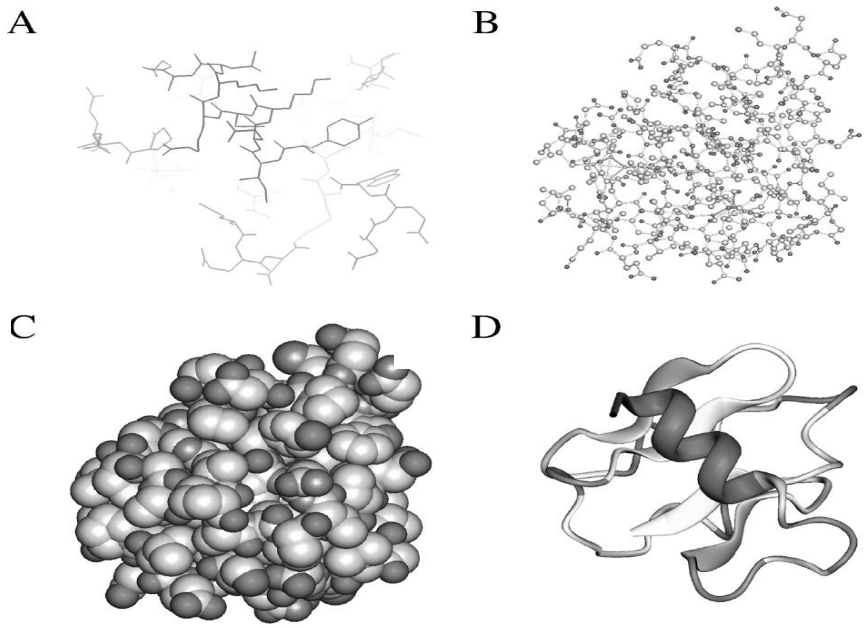
Најчесто користени молекуларни визуелизациски формати за приказ на структурата на протеините се следниве:

- *Молекуларен визуелизациски формат што се задржува само на приказ на врските помеѓу атомите, прикажувајќи ги како отсечки.*

- *Молекуларен визуелизациски формат што ја прикажува протеинската молекула како множество на атоми, претставени со сфери, каде врските помеѓу атомите се претставени со отсечки.*

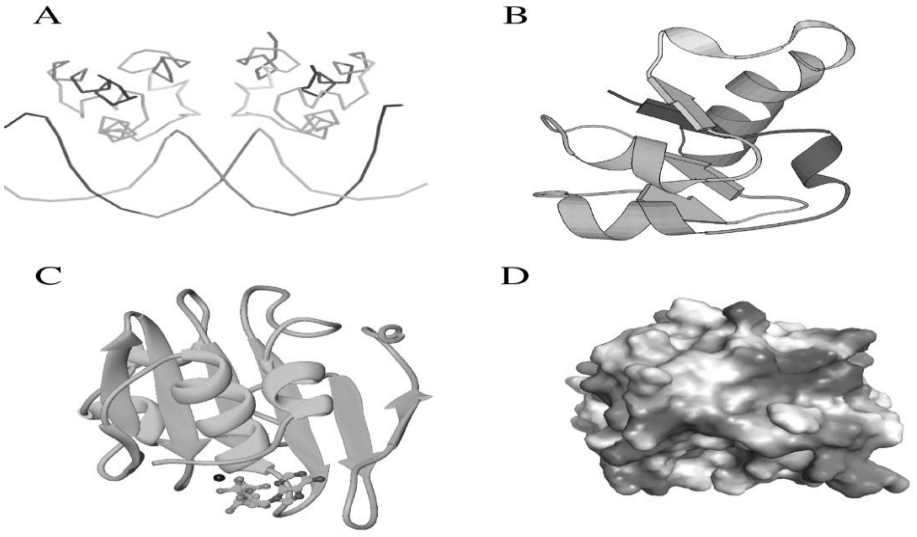
- *Молекуларен визуелизациски формат што ја прикажува протеинската молекула како множество на атоми со ван дер валсови радиуси, со пропорционална распределба во рамки на приказот.*

- *Молекуларен визуелизациски формат што ја прикажува просторната распределба на секундарните извиткувања во рамки на протеинската молекула.*



Слика 6.3.1. Приказ на четирите молекуларни визуелизациски формати

Најчесто употребувани апликации за приказ на молекуларната структура на протеините се: *RasMol*, *Swiss-PDBViewer*, *Molscript*, *Ribbons*, *Grasp*,…итн. Множеството на координати на атоми претставува влез, додека излез е приказот на молекуларната структура на протеинот. Различноста на молекуларниот приказ кај: Rasmol, Molscript, Ribbons и Grasp е дадена на Слика 6.3.2.

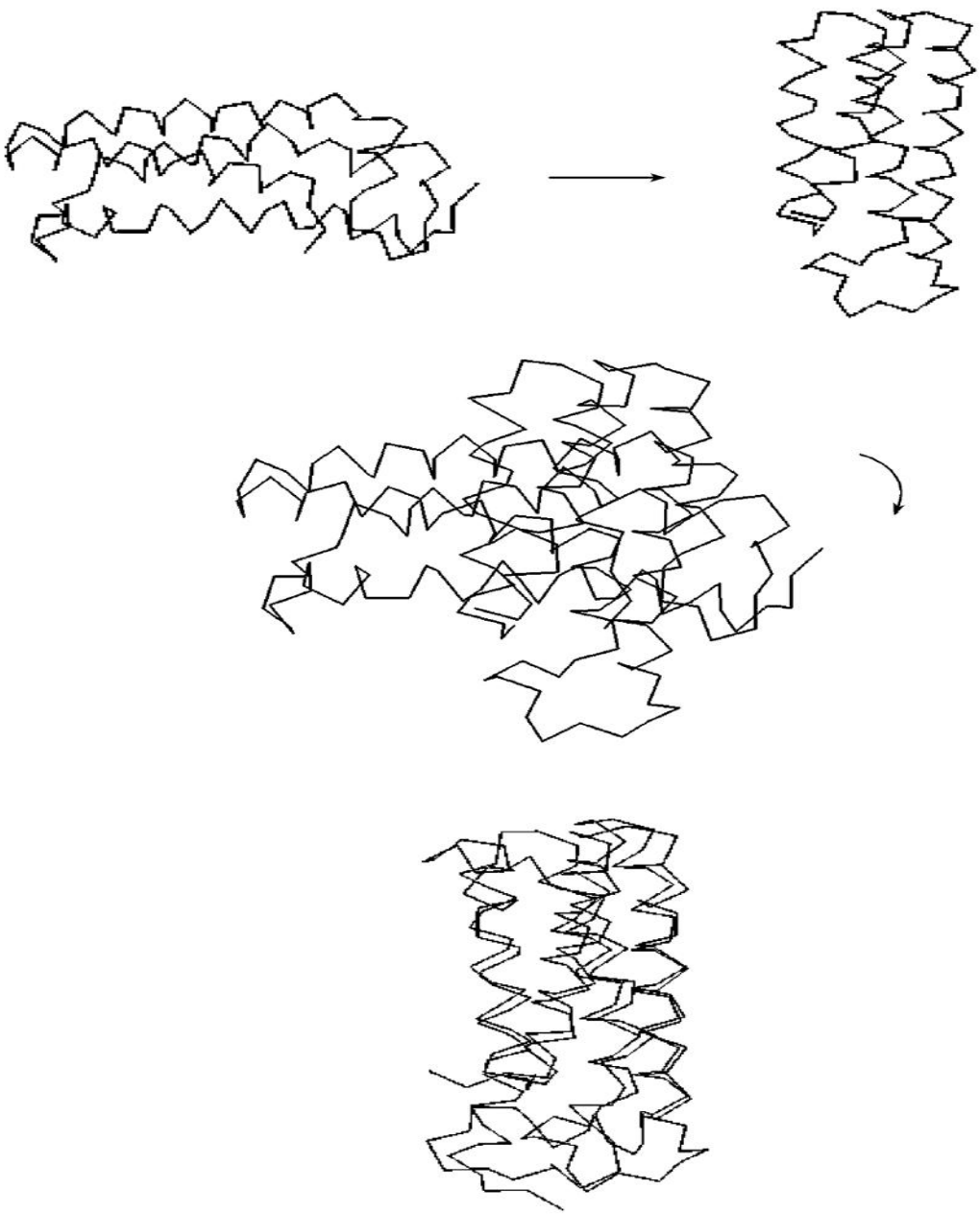


Слика 6.3.2. Молекуларен приказ кај: (A) Rasmol, (B) Molscript, (C) Ribbons и (D) Grasp

6.4. Споредување на протеини

За да се утврди сличноста помеѓу два протеини истите се споредуваат. Процесот на споредба вклучува порамнување на примарните структури и просторна споредба на извиткувањето кај протеините. Во случај на подалечна еволуциска врска, можно е два протеини да имат слична просторна структура и покрај различноста на нивните примарни структури. Просторната споредба на протеините се врши алгоритамски, со примена на интермолекуларниот или интрамолекуларниот метод, а понекогаш се употребува и хибриден метод.

Интермолекуларниот метод се применува за споредба на слични протеини. По утврдување на еквивалентни резидиумски парови, со трнаслација на еден од протеините, истите се доведуваат во заедничка координатна рамка. Откако протеините ќе се доведат во иста координатна рамка, транслираниот протеин се ротира во однос на референтниот протеин, при што постојано се мерат растојанијата помеѓу централните јаглеродни атоми C_α кај еквивалентните резидумски парови. Ротацијата завршува кога меѓу-молекуларното растојание е минимално, што соодветствува на минимум на функцијата $f = \sqrt{\frac{\sum_{i=1}^{n} D_i^2}{N}}$, каде N е број на еквивалентни резидумски парови, додека со D_i се означени растојанијата помеѓу централните јаглеродни атоми C_α. Просторната положба на протеините, за која меѓу-молекуларното растојание е минимално, одговара на максимално просторно совпаѓање помеѓу протеинските молекули.



Слика 6.4.1. Интермолекуларен метод за просторна споредба на протеински молекули

За разлика од интермолекуларниот метод, интрамолекуларниот метод се применува за споредба на протеини со различна примарна структура. Методот се базира на конструкција на матрица на резидиумски растојанија за секој протеини, одделно. Со споредба на матриците на резидумски растојанија може да се утврди максимално меѓусебно совпаѓање.

6.5. Класификација на протеини

За да се класифицира протеин, истиот претходно се споредува. Класификацијата на протеини воспоставува хиерархиска релација помеѓу протеините и истата овозможува еволуциски преглед на развојот на протеинските структури. Во денешни услови, протеините се класифицираат софтверски, при што два најпознати системи за класификација на протеини се системите: *SCOP* и *CATH*.

SCOP(Structural Classsifiaction Of Proteins) овозможува хиерархиска организација на протеините во: класи, класи на извиткување, суперфамилии и фамилии. Фамилиите кај SCOP вклучуваат протеини со слична примарна структура (повеќе од 30% идентичност на примарно ниво). Членовите на суперфамилиите имат далечен заеднички предок и истите функционално се разликуваат. Множество на супермаилии со слични секундарни изиткувања, пропратени со слични просторни ориентации и врски формира извиткувачка класа. Класификацијата на протеини по класи, ги групира протеините по секундарни извиткувања, при што постои: класа на протеини која вклучува само алфа хелиски, класа на протеини која вклучува само бета рамнини и класа на протеини која вклучува алфа хеликси и бета рамнини,..итн.

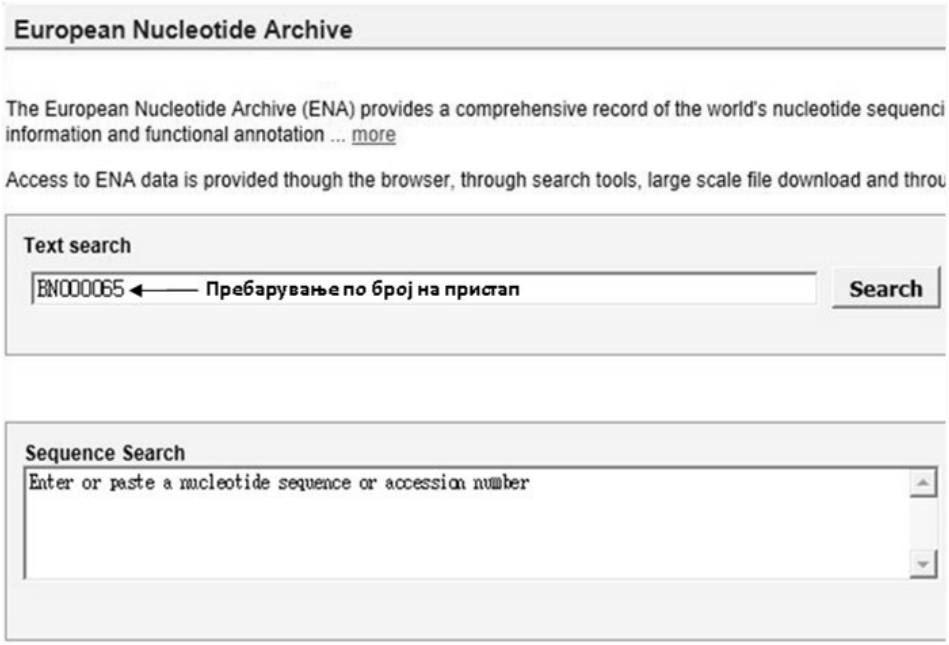
Кај CATH(Class, Architecture, Topology and Homologous) системот, протеините се организирани во класи, архитектури, топологии, хомологни суперфамилии и хомологни фамилии. Фамилиите и суперфамилиите се дефинирани на идентичен начин како кај SCOP. Она што претставува извиткувачка класа кај SCOP, кај CATH е топологија. Архитектурата ја опишува распределбата на секундарните извиткувања, независно од меѓусебната поврзаност.

7. Додаток – Работа со Архива на нуклеотидни скевенци (ENA)

ENA (Европска нуклеотидна архива) се состои од три бази на податоци: *ЕМБЛ-Банка (EMBL-Bank)*, *Архива на исчитувања на секвенци (Sequence Read Archive)* и *Архива на траги (Trace Archive)*. Европската Нуклеотидна Архива овозможува пристап до целосни и парцијални исчитувања на нуклеотидни секвенци. ЕМБЛ-Банката ги архивира целосните исчитувања, додека парцијалните исчитувања се додаваат во Архивата на исчитувања на секвенци. Целосните исчитувања се добиваат со порамнување и соединување на кратки – претходно секвенционирани сегменти. Некои региони кај целосните исчитувања се биолошки означени како: егзони, интрони или гени. Податоците добиени со примена на некоја од методите за секвенционирање од наредната генерација се додаваат во Архивата на исчитувања, додека во Архивата на траги се додаваат податоци за капиларно секвенционирани фрагменти.

Европската Нуклеотидна Архива прифаќа податоци за целосни и парцијални исчитувања на нуклеотидни секвенци, независно од биолошката означеност. Податоците се доставуваат од независни истражувачи, секвенционирачки конзорциуми и патент канцеларии. Доставените податоци не се филтрират, што значи дека може да постојат дупликат записи. Ажурирањето на записот е право и привилегија на авторот (доставувачот) на податоците за секвенцата.

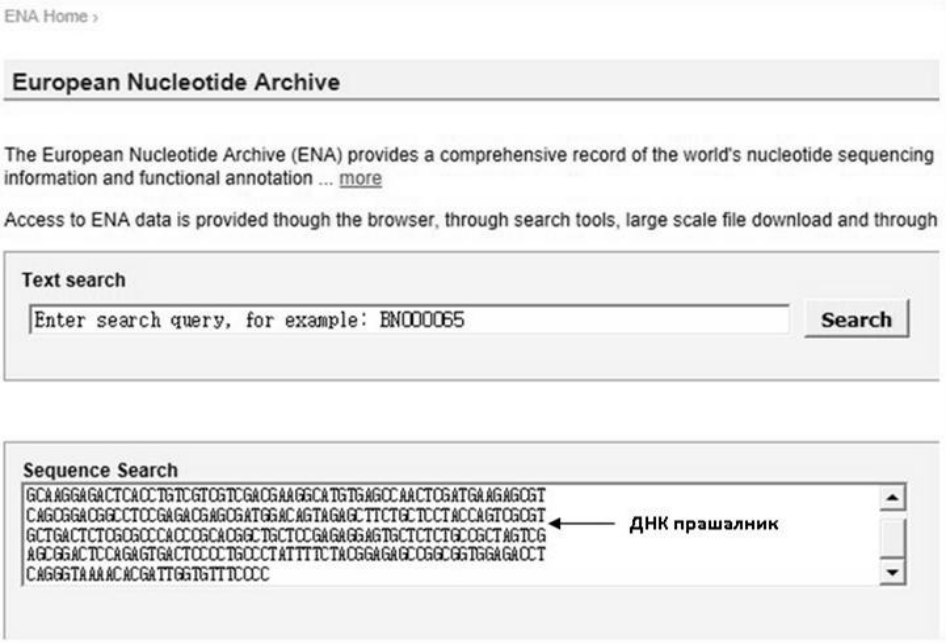
Европската Нуклеотидна Архива овозможува пребарување по текст и секвенца, Слика 7.1. Текстуалното пребарување на Европската Нуклеотидна Архива овозможува пребарување по: име на ген, име на болест, број на пристап, клучен термин, податочна класа и таксономска поделба. Во случај на пребарување по број на пристап, Слика 7.1, резултатот од пребарувањето соодветствува на бараниот запис. Резултатот од пребарувањето е листа на записи, кога пребарувањето се врши по клучен термин, Слика 7.2.



Слика 7.1. Почетна страна на ЕНА

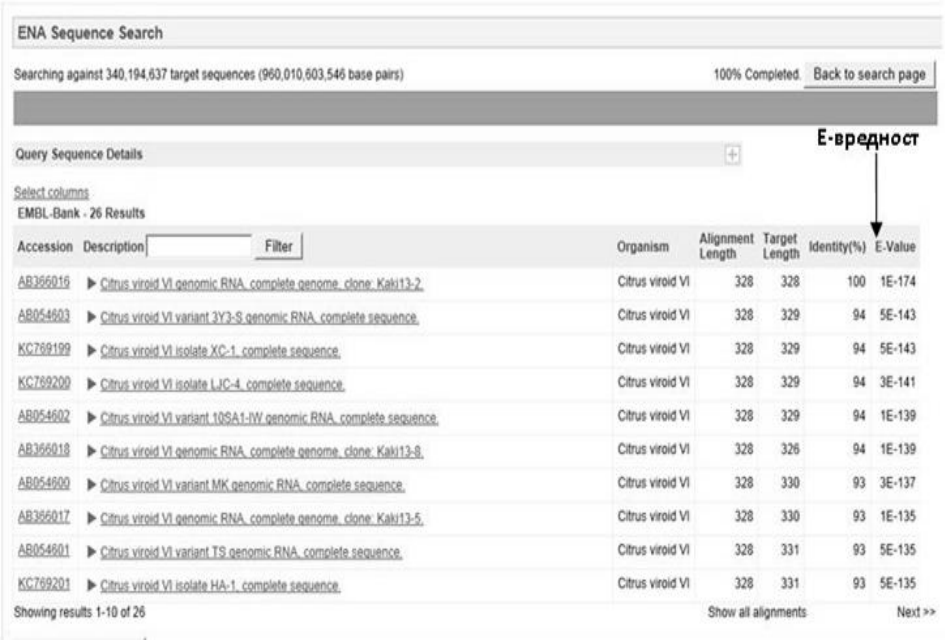
Слика 7.2. Пребарување по клучен термин ,,Mouse”

За разлика од Архивата на исчитувања на секвенци и Архивата на траги, кои се непогодни за пребарување по секвенца, ЕМБЛ-Банката може да се пребарува по секвенца, Слика 7.3. ENA пребарувачот овозможува пребарување по ДНК или РНК прашалници. Бројот на пристап или редоследот на нуклеотиди, се влезни параметри при пребарување по секвенца.



Слика 7.3. Пребарување по секвенца

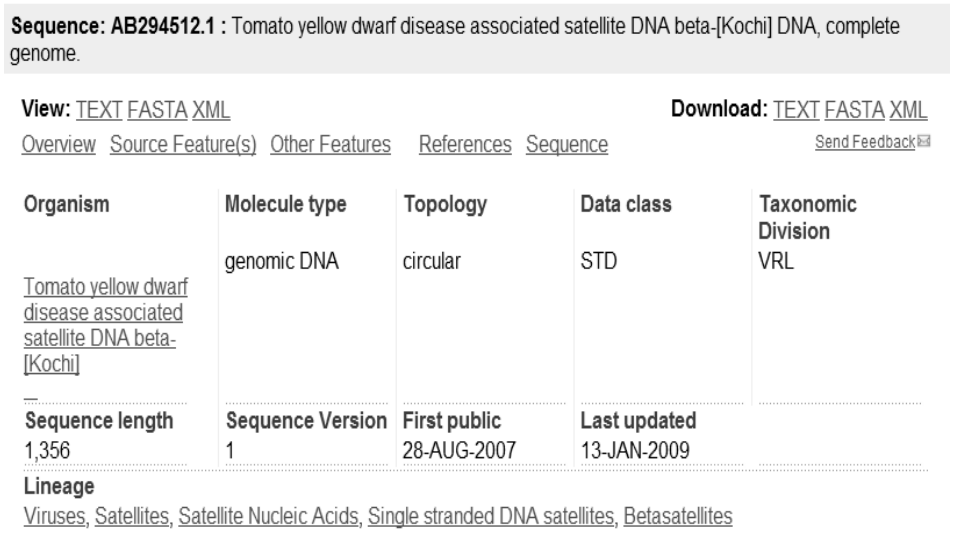
Резултатите од пребарувањето се подредени по E-Вредност (проценка на значење на совпаѓање), започнувајќи од најмалата. Најмала E-вредност соодветствува на најзначаен резултат, Слика 7.4. Се пресметуваат и должина на порамнување и процент на идентичност, Слика 7.4. Должина на порамнување е должина на совпаѓање помеѓу целната и прашалник секвенцата. Процентот на идентичност е процент на идентични нуклеотиди помеѓу целната и прашалник секвенцата.



Слика 7.4. Резултат од пребарување по ДНК прашалник

ЕНА страницата за пребарување на сличност (ENA Sequence Similarity Page http://www.ebi.ac.uk/Tools/sss/), обезбедува додатни опции за пребарување. Овие опции се имплементации на познати алгоритми за пребарување, како: BLAST, PSI-BLAST, FASTA и SSEARCH. Истите се употребуваат при пребарување на: дел од ЕМБЛ- Банката, пребарување на други бази на податоци и пребарување по кратки прашалник секвенци.

Пристапниот број, описот на записот, опциите за преглед и преземање на содржината на записот во FASTA, XML или текстуален облик, како и деталите за секвенцата: потекло на секвенцата, должината на секвенцата, типот и топологијата на молекулата, верзијата на секвенца и датумите на прва објава и последна промена на содржината на записот се општи податоци, кои се наведени на почеток на запис од ЕМБЛ-Банката, Слика 7.5.

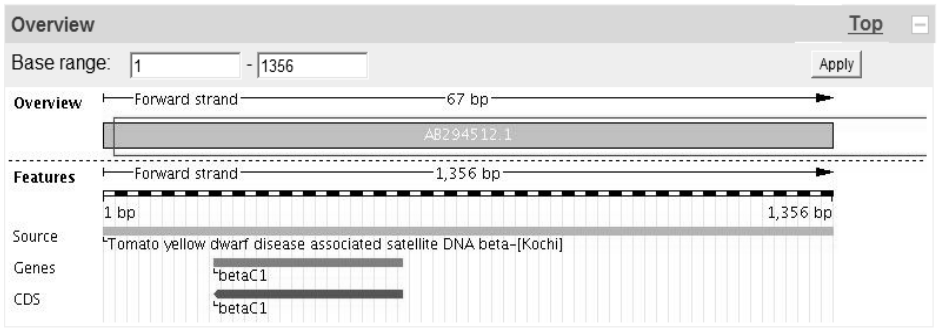


Слика 7.5. Запис од ЕМБЛ-Банка

Референците до други бази на податоци, каде се наведени додатни и корисни информации за секвенцата, како на пример додатна означеност на секвенцата, се наоѓаат во навигацискиот панел, Слика 7.6.

Слика 7.6. Навигациски панел

Преглед секцијата врши графички приказ на биолошки означените особини кај секвенцата, како: и-РНК, гени, интрони и егзони. Овие податоци ги доставува авторот на секвенцата, Слика 7.7.



Слика 7.7. Секција ,,Преглед"

Секцијата ,,Потекло" интегрира податоци за потеклото на секвенцата, Слика 7.8.

Source Feature(s)	Top
Source(s)	
Taxon:	Taxon:427315
source	**1..1356**
organism	Tomato yellow dwarf disease associated satellite DNA beta-[Kochi]
country	Japan:Kochi, Takaoka-gun
isolation source	tomato with yellow dwarf symptoms
collection date	2000
clone	pTKbeta-1

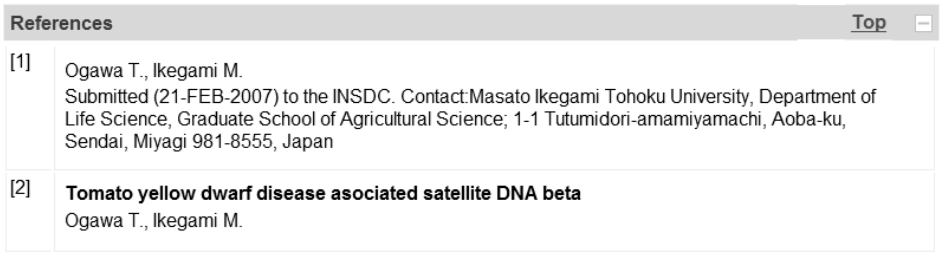
Слика 7.8. Секција ,,Потекло"

Секцијата ,,Останати Својства" овозможува детален преглед на биолошки означените својства. Таму се наведени: локациите на егзоните, гените, резултатот од преводот, како и примарната структура на протеинот, Слика 7.9.

Other Features	Top
Base range: 1 - 1356 Show main features only	Apply
Showing results 1 - 2	
CDS	**complement(206..556)**
codon_start	1
transl_table	1
gene	betaC1
product	betaC1 protein
translation	MTITYNNGKGIKFIVDVRLHQLLKVIVQVYSTNKPVLIGFKCHIPYTYVQMVPPFDFNGAEELIRETLELMYEDSDISNF KQEEMIDSIDMVMMHWLGHMGVDTVDRYTIRCRNTV
Coding:	BAF75929
InterPro	IPR018583
UniProtKB/TrEMBL	A7M6K4

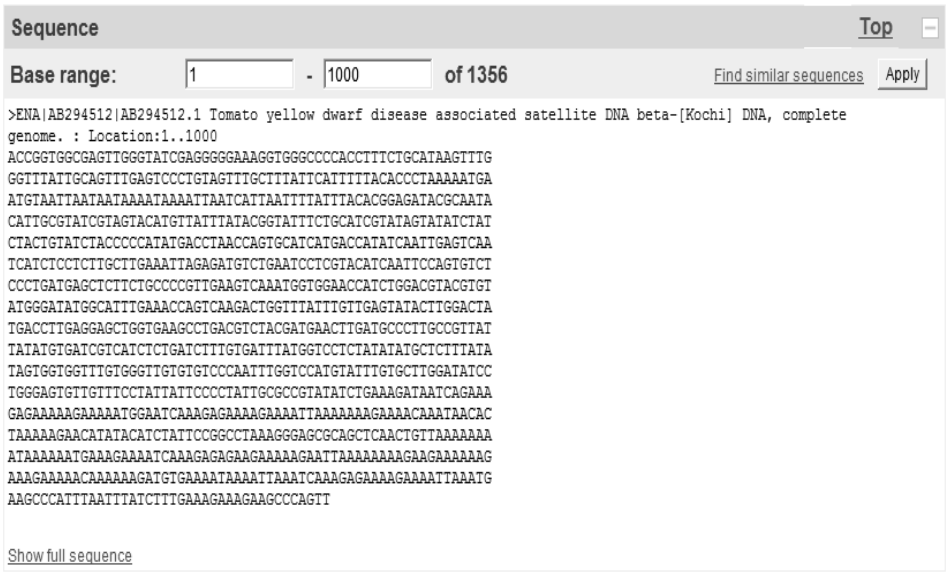
Слика 7.9. Секција ,,Останати Својства”

Референците, како означеност на секвенцата од друга страна, се наведени кон крајот на записот, Слика 7.10.



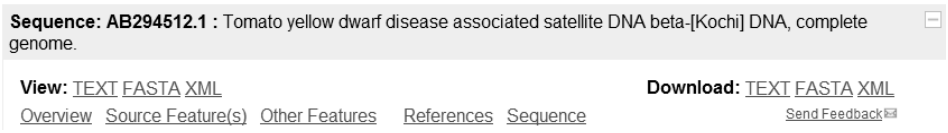
Слика 7.10. Приказ на референци

Парцијалниот или целосниот приказ на редоследот на нуклеотидите во FASTA формат е прикажан на крајот од записот, Слика 7.11.



Слика 7.11. Редослед на нуклеотиди

Европската Нуклеотидна Архива овозможува преземање на еден или повеќе записи, Слика 7.12. Откако ќе се пронајде бараниот запис, истиот може да биде превземен во: FASTA, XML или текстуален облик. Со пребарување на ЕНА по клучен термин, постои можност за преземање на резултатите од пребарување во претходно наведените податочни облици.



Слика 7.12. Преземање на секвенца

Литература:

[**1**] Anthea, M., Hopkins, J., McLaughlin, C., Johnson, S., Warner, M., LaHart, D. & Wright, J. (1993). Human Biology and Health. Prentice Hall.

[**2**] Altschul, S., Gish, W., Miller, W., Myers, E. & Lipman, D. (1990). Basic local alignment search tool. Journal of Molecular Biology, 215(3), 403–410.

[**3**] Arabidopsis, Genome Initiative. (2000). Analysis of the genome sequence of the flowering plant Arabidopsis thaliana. Nature 408(6814), 796.

[**4**] Benson, A., Ilene, K., Lipman, D., Ostell, J. & Sayers, E. (2010). GenBank. Nucleic acids research, 38(1), D46-D51.

[**5**] Benson, D., Karsch-Mizrachi, I., Lipman, J., Ostell, J. & Sayers, E. W. (2009). GenBank. Nucleic Acids Research, 37 (Database), D26–D31.

[**6**] Benson, D., Karsch-Mizrachi, I., Lipman, J., Ostell, J. & Wheeler, D. (2008). GenBank. Nucleic Acids Research, 36 (Database), D25–D30.

[**7**] Blattner, R. (1997). The complete genome sequence of Escherichia coli K-12. Science 277(5331), 1453-1462.

[**8**] Bray, N., Dubchak, I. & Pachter, L. (2003). AVID: A global alignment program. Genome research, 13(1), 97-102.

[**9**] Bucher, P. (1990). Weight matrix descriptions of four eukaryotic RNA polymerase II promoter elements derived from 502 unrelated promoter sequences. J. Mol. Biol, 212, 563–578.

[**10**] Burkhard, M., Dress, A. & Werner, T. (1996). Multiple DNA and protein sequence alignment based on segment-to-segment comparison. Proceedings of the National Academy of Sciences 93, 22, 12098-12103.

[**11**] Burkhard, M., Frech, K., Dress, A. & Werner, T. (1998). DIALIGN: finding local similarities by multiple sequence alignment. Bioinformatics, 14(3), 290-294.

[**12**] Chou, PY. & Fasman, GD. (1974). Prediction of protein conformation. Biochemistry, 13(2), 222–245.

[**13**] Chou, PY. & Fasman, GD. (1978). Empirical predictions of protein conformation. Annu Rev Biochem, 47, 251–276.

[**14**] Chou, PY. & Fasman, GD. (1978). Prediction of the secondary structure of proteins from their amino acid sequence. Adv Enzymol Relat Areas Mol Biol, 47, 145–148.

[**15**] Crick, F. (1970). Central dogma of molecular biology. Nature, 227(5258), 561-563.

[**16**] Di Lullo, G., Sweeney, M., Körkkö, J., Ala-Kokko, L. & James, D. (2002). Mapping the Ligand-binding Sites and Disease-associated Mutations on the Most Abundant Protein in the Human, Type I Collagen. J. Biol. Chem, 277 (6), 4223–4231.

[**17**] Elgar, G. & Vavouri, T. (2008). Tuning in to the signals: noncoding sequence conservation in vertebrate genomes. Trends Genet., 24(7), 344–352.

[**18**] Ficket, W. (1984). Fast optimal alignment. Nucleic Acids Res., 12(1), 175-179.

[**19**] Ford, C.E & Hamerton, J.L. (1956). The chromosomes of Man. Nature, 178(4541), 1020-1023.

[**20**] Garnier, J., Gibrat, JF. & Robson, B. (1996). GOR method for predicting protein secondary structure from amino acid sequence. Methods Enzymol, 266, 540-553.

[**21**] Gotoh, O. (1982). An improved algorithm for matching biological sequences. Journal of Molecular Biology, 162(3), 705–708.

[**22**] Gusfield, D. (1997). Algorithms on strings, trees and sequences: computer science and computational biology. Cambridge University Press.

[**23**] Guy, C., Charles, E. & Birney, E. (2012). The future of DNA sequence archiving. GigaScience, 1(1), 2.

[**24**] Ha, SC., Lowenhaupt, K., Rich, A., Kim, YG. & Kim, KK. (2005). Crystal structure of a junction between B-DNA and Z-DNA reveals two extruded bases. Nature, 437(7062), 1183–1186.

[**25**] Harley, C.B. & Reynolds, R.P. (1987). Analysis of E.coli promoter sequences. Nucleic Acids Res, 15, 2343–2361.

[**26**] Hirschberg, D. (1975). A linear space algorithm for computing maximal common subsequences. Communications of the ACM, 18(6), 341–343.

[**27**] Ho, PS. (1994). The non-B-DNA structure of d(CA/TG)n does not differ from that of Z-DNA. Proc Natl Acad Sci USA, 91(20), 9549–9553.

[**28**] International Human Genome Sequencing Consortium., (2004). Finishing the euchromatic sequence of the human genome. Nature 431(7011), 931-945.

[**29**] Konstantinos, G., Papanikolaou, G. & Pantopoulos, K. (2012). Regulation of iron transport and the role of transferrin. Biochimica et Biophysica Acta (BBA)-General Subjects, 1820(3), 188-202.

[**30**] Lipman, D. & Pearson, W. (1985). Rapid and sensitive protein similarity searches. Science, 227(4693), 1435–1441.

[**31**] Maxam, AM. & Gilbert, W. (1977). A new method for sequencing DNA. Proc. Natl. Acad. Sci. U.S.A, 74(2), 560–564.

[**32**] Myers, E. & Miller, W. (1988). Optimal alignments in linear space. Computer Applicationsin the Biosciences, 4, 11–17.

[**33**] Needleman, S. & Wunsch, C. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. Journal of molecular biology, 48(3), 443-453.

[**34**] Notredame, C., Holme, L., Heringa, J., Suhre, K. & Abergel, C. (2010). Tcoffee©: Multipurpose sequence alignments program. Journal of Cell and Molecular Biology, 71.

[**35**] Perier, R., Praz, V., Junier, T., Bonnard, C. & Bucher, P. (2000). The eukaryotic promoter database (EPD). Nucleic Acids Research, 28(1), 302-303.

[**36**] Praz, V., Perier, R., Bonnard, C. & Bucher, P. (2002). The eukaryotic promoter database, EPD: new entry types and links to gene expression data. Nucleic Acids Research, 30(1), 322-324.

[**37**] Rasko, L., Akhtar, R., Birney, E., Bower, L., Cerdeno-Tárraga, A., Cheng, Y. & Cleland, I. (2011). The European nucleotide archive. Nucleic acids research, 39(1), D28-D31.

[**38**] Rich, A., Norheim, A., Wang, AHJ. (1984). The chemistry and biology of left-handed Z-DNA. Annual Review of Biochemistry, 53(1), 791–846.

[**39**] Sanger, F. & Coulson, AR. (1975). A rapid method for determining sequences in DNA by primed synthesis with DNA polymerase. J. Mol. Biol, 94(3), 441–448.

[**40**] Schmid, C., Praz, V., Delorenzi, M., Perier, R. & Bucher, P. (2004). The eukaryotic promoter database EPD: the impact of in silico primer extension. Nucleic Acids Research, 32 Database Issue, D82-D85.

[**41**] Shen, S., Yang, J., Yao, A. & Hwang, P. (2002). Super pairwise alignment (SPA): an efficient approach to global alignment for homologous sequences. Journal of Computational Biology, 9(3), 477-486.

[**42**] Sinden, R. (1994). DNA structure and function. Academic Press, 398.

[**43**] Smith, F. & Waterman, M. (1981). Identification of common molecular subsequences. Journal of molecular biology, 147(1), 195-197.

[**44**] Stojanov, D., Koceski, S. & Mileva, A. (2013). FLAG: Fast Local Alignment Generating Methodology. Romanian Biotechnological Letters, 18(1), 7881-7888.

[**45**] Stojanov, D., Mileva, A. & Koceski, S. (2012). A new, space-efficient local pairwise alignment methodology. Advanced Studies in Biology, 4(2), 85-93.

[**46**] Tateno, Y., Fukami-Kobayashi, K., Miyazaki, S., Sugawara, H. & Gojobori, T. (1998). DNA Data Bank of Japan at work on genome sequence data. Nucleic Acids Res., 26, 16–20.

[**47**] Temin, HM. & Mizutani, S. (1970). RNA-dependent DNA polymerase in virions of Rous sarcoma virus. Nature, 226(5252), 1211–1213.

[**48**] Thompson, J., Higgins, D. & Gibson, T. (1994). CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. Nucleic acids research, 22(22), 4673-4680.

[**49**] Tjio, JH. & Levan, A. (1956). The chromosome of number of Man. Hereditas, 42, 1-6.

[**50**] Travaglini-Allocatelli, C., Ivarsson, Y., Jemth, P. & Gianni, S. (2009). Folding and stability of globular proteins and implications for function. Curr Opin Struct Biol, 19(1), 3–7.

[**51**] Turnpenny, P. & Ellard, S. (2005). Emery's Elements of Medical Genetics, 12th. ed. Elsevier, London.

[**52**] Wang, JC. (1979). Helical repeat of DNA in solution. PNAS, 76(1), 200–203.

[**53**] Watson, JD. & Crick, FH. (1953). Molecular structure of nucleic acids; a structure for deoxyribose nucleic acid. Nature, 171(4356), 737-738.

[**54**] Yoshio, T., Imanishi, T., Miyazaki, S., Fukami-Kobayashi, K., Saitou, N., Sugawara, H. & Gojobori, T. (2002). DNA Data Bank of Japan (DDBJ) for genome scale research in life science. Nucleic Acids Research, 30(1), 27-30.